

Cortex XSOAR Administrator's Guide

5.5

Contact Information

Corporate Headquarters:

Palo Alto Networks

3000 Tannery Way

Santa Clara, CA 95054

www.paloaltonetworks.com/company/contact-support

About the Documentation

- For the most recent version of this guide or for access to related documentation, visit the Technical Documentation portal www.paloaltonetworks.com/documentation.
- To search for a specific topic, go to our search page www.paloaltonetworks.com/documentation/document-search.html.
- Have feedback or questions for us? Leave a comment on any page in the portal, or write to us at documentation@paloaltonetworks.com.

Copyright

Palo Alto Networks, Inc.

www.paloaltonetworks.com

© 2020-2020 Palo Alto Networks, Inc. Palo Alto Networks is a registered trademark of Palo Alto Networks. A list of our trademarks can be found at www.paloaltonetworks.com/company/trademarks.html. All other marks mentioned herein may be trademarks of their respective companies.

Last Revised

May 5, 2020

Table of Contents

Cortex XSOAR Overview.....	11
Cortex XSOAR Licenses.....	13
Cortex XSOAR License Types.....	13
Cortex XSOAR Users.....	13
Add a License.....	13
Product Support Lifecycle.....	15
Cortex XSOAR Telemetry.....	17
Data Usage Collection.....	17
Runtime Data Usage Collection.....	19
Cortex XSOAR Concepts.....	20
Incidents.....	20
Incident Fields.....	20
Incident Lifecycle.....	20
Integrations.....	20
Integration Instance.....	21
Playbooks.....	21
Automations.....	21
Commands.....	21
War Room.....	21
Indicators and Indicator Types.....	21
Playground.....	22
Jobs.....	22
Use Cases.....	23
Analytics and SIEM.....	23
Authentication.....	23
Case Management.....	23
Data Enrichment & Threat Intelligence.....	23
Email Gateway.....	24
Endpoint.....	24
Forensics and Malware Analysis.....	24
Network Security (Firewall).....	25
Network Security (IDS/IPS).....	25
Vulnerability Management.....	25
Keyboard Shortcuts.....	26
Playbooks, Scripts, and CLI.....	26
Open and Close Right Shoulders.....	27
How to Search in Cortex XSOAR.....	28
Using the Search Query.....	28
Configure System Notifications.....	30
Install DBot for Slack.....	31
Single Server Deployment.....	33
System Requirements.....	35
Cortex XSOAR Server.....	35
Cortex XSOAR Engine.....	35
Web Browsers.....	36
Required URLs.....	36
Performance Benchmark.....	38
Disk Usage.....	38

Performance Benchmark Test.....	38
Benchmark Process.....	39
Benchmark Results.....	39
Install Cortex XSOAR for a Single Server Deployment.....	41
Installer Flags.....	42
Install Cortex XSOAR Offline.....	46
Dependencies for Offline Installation.....	46
Post-Installation Checklist.....	48
Server Post-Installation Health Check.....	48
Monitor Cortex XSOAR Components.....	49
HTTPS with a Signed Certificate.....	50
AWS EC2 Deployment Guidelines.....	53
Upgrade the Cortex XSOAR Server.....	55
Uninstall Cortex XSOAR.....	56

Distributed Database Deployment.....57

Distributed Database Deployment.....	59
Single database server.....	59
Multiple database servers.....	60
Sizing Requirements for Distributed Database Deployment.....	61
Install Cortex XSOAR for a Distributed Database Deployment.....	62
Install a Distributed Database Node.....	64
Configure a Live Backup for a Distributed Database Overview.....	65
Configure the Live Backup Environment for a Distributed Database.....	65
Transition a Standby Server to Active Mode.....	68
Transition an Active Server to Standby Mode for a Distributed Database.....	69
Change the Node Admin Password.....	70
Delete a User from a Node.....	71
Convert a Single Server Deployment to a Distributed Database Deployment.....	72
Reindex Databases in a Distributed Database Deployment.....	74
Restore Databases in a Distributed Database Deployment.....	75
Upgrade the Cortex XSOAR Server for a Distributed Database.....	76

Proxy..... 77

Configure Proxy Settings.....	79
Use NGINX as a Reverse Proxy to the Cortex XSOAR Server.....	80
Use Engines Through the NGINX Reverse Proxy.....	80
Install NGINX on Cortex XSOAR.....	80
Generate a Certificate for NGINX.....	80
Configure NGINX.....	81

Manage Data.....83

Reindex the Entire Database.....	85
Reindex a Specific Index Database.....	86
Reindex the Entire Database for a Distributed Database.....	87
Reindex a Specific Index for a Distributed Database.....	88
Free up Disk Space with Data Archiving.....	89
Migrate Data to Another Server.....	91
Move Data Folders to Another Location on the Server.....	92
Restore an Archived Folder.....	93

Users and Roles	95
Users and Roles Overview.....	97
Roles in Cortex XSOAR.....	98
Define a Role.....	98
Default Admin.....	101
Self-Service Read-Only Users.....	102
Configure the Server for Self Service Read-Only Users.....	102
Create the Self Service Read-Only Users.....	103
Create the Read-Only Dashboard.....	103
Create the Read-Only Incident Type and Layout.....	104
User Settings and Preferences.....	105
Messages.....	105
Details.....	105
Preferences.....	105
Notifications.....	106
Shift Management.....	107
Managing Shifts.....	107
User Invitations.....	109
Invite a User.....	109
Integration Permissions.....	111
Password Policy.....	112
Create a Password Policy.....	112
Edit a Default Password Policy.....	112
Change the Administrator Password.....	115
Authenticate Users with SAML 2.0.....	116
Set up Okta as the Identity Provider Using SAML 2.0.....	116
Set up Microsoft Azure as the Identity Provider.....	124
Set up ADFS as the Identity Provider Using SAML 2.0.....	131
Configure User Notifications.....	144
Set the Default Theme for New Users.....	145
Disaster Recovery and Live Backup	147
Disaster Recovery and Live Backup Overview.....	149
Host Names, DNS, and Disaster Recovery.....	150
Configure the Live Backup Environment.....	151
Configure Live Backup for Multiple SAMLs.....	152
DR Scenario #1: Testing the DR Environment.....	153
DR Scenario #2: Unrecoverable Active Server Failure.....	154
DR Scenario #3: Unrecoverable Standby Server Failure.....	154
Transition an Active Server to Standby Mode.....	156
Transition a Standby Server to Active Mode.....	157
Transition Between DR States Through the Configuration File.....	158
Upgrade the Live Backup Environment.....	159
Cortex XSOAR Engines and Disaster Recovery.....	160
Backup the Database.....	161
Remote Repositories in Cortex XSOAR	163
Remote Repositories Overview.....	165
Configure a Remote Repository on a Development Machine.....	166
Configure a Remote Repository on the Production Machine.....	169
Edit and Push Content to a Remote Repository.....	171

Troubleshoot a Remote Repository Configuration.....	173
Troubleshoot a Remote Repository Definition.....	173
Troubleshoot Editing and Pushing Content.....	174
Troubleshoot Content Issues.....	175
Engines.....	177
Cortex XSOAR Engines Overview.....	179
Engine Proxy.....	179
Engine Architecture.....	179
Engine Load-Balancing.....	179
Engine Installation and Configuration.....	180
Install Cortex XSOAR Engines.....	181
Run the Engine as a Service on Windows.....	183
Use an Engine in an Integration.....	185
Manage Engines.....	186
Configure Engines.....	187
Edit the Engine Configuration.....	187
Configure the Engine to Use a Web Proxy.....	188
Configure the Engine to Call the Server Without Using a Proxy.....	189
Configure the Number of Workers for the Server and Engine.....	189
Configure Access to Communication Tasks through an Engine.....	190
Notify Users When an Engine Disconnects.....	190
Remove the Cortex XSOAR Server From the Load-Balancing Group.....	190
Remove an Engine.....	191
Troubleshoot Cortex XSOAR Engines.....	192
Troubleshoot Engine Upgrades.....	192
Docker.....	193
Docker Installation.....	195
Install Docker Enterprise Edition on Cortex XSOAR.....	195
Install Docker Community Edition on Cortex XSOAR.....	196
Install Docker Distribution for Red Hat on Cortex XSOAR.....	197
Install Docker Images Offline.....	199
Configure Python Docker Integrations to Trust Custom Certificates.....	200
Docker Images in Cortex XSOAR.....	202
Manage Docker Images.....	203
Create a Docker Image In Cortex XSOAR.....	204
Docker Hardening Guide.....	206
Configure Memory Limit Support Without Swap Limit Capabilities.....	208
Run Docker with Non-Root Internal Users.....	208
Use a Docker Image for Python Scripts.....	209
Configure the Memory Limitation.....	209
Test the Memory Limit.....	210
Limit Available CPU.....	211
Configure the PIDs Limit.....	211
Configure the Open File Descriptors Limit.....	212
Troubleshoot Docker Networking Issues.....	213
Run Docker with Non-Root Internal Users.....	214
Dashboards.....	217
Dashboard Overview.....	219
Create a Dashboard.....	220

Add a Widget to a Dashboard.....	221
Configure a Default Dashboard.....	222
Share and Unshare a Dashboard.....	223
Edit a Dashboard.....	224
Reports.....	225
Reports Overview.....	227
Chromium Installation for Reports.....	228
Install Chromium on Fedora, RHEL, or CentOS.....	228
Install Chromium on openSUSE and SUSE.....	228
Install Chromium on Ubuntu or Debian.....	228
Configure Cortex XSOAR to Use PhantomJS.....	230
Create a Report.....	231
Schedule a report.....	232
Schedule a Report Examples.....	232
Create an Incident Summary Report.....	235
Add a Widget to a Report.....	236
Edit a report.....	237
Change the Report Logo.....	238
Configure the Time Zone and Format in a Report.....	239
Troubleshoot Reports.....	240
Widgets.....	241
Widgets Overview.....	243
Create a Widget in the Widgets Library.....	244
Widget Parameters.....	244
Create a Custom Widget Using a JSON File.....	246
JSON File Widget Parameters.....	246
JSON File Widget Example.....	248
Create a Custom Widget Using an Automation Script.....	250
Script Based Widgets Using Automation Scripts Examples.....	250
Create a Widget from an Indicator.....	258
Edit a Widget.....	260
Create a Used Percentage Widget for a Disk Partition.....	261
Saved By Dbot (ROI) Widget.....	263
Customize the Currency Symbol in the Saved by Dbot Widget.....	264
Manage Indicators.....	265
Understand Indicators.....	267
Feed Integrations.....	267
Indicators Page.....	269
Indicator Reputation.....	270
Indicator Types.....	272
Indicator Fields.....	275
Exclusion List.....	277
Create a Feed-Triggered Job.....	277
Manage the Indicator Timeline.....	278
Auto Extract Indicators.....	279
Auto Extract Modes.....	279
How to Define Auto Extract.....	280
Configure What Auto Extract Executes.....	280
Disable Auto Extract for Scripts and Integrations.....	281

Auto Extract Indicators from a Phishing Email.....	281
--	-----

Incidents.....	285
-----------------------	------------

Incident Lifecycle.....	287
Planning.....	287
Configure Integrations.....	288
Classification Mapping.....	288
Pre-Processing.....	288
Incident Created.....	288
Running Playbooks.....	288
Post-Processing.....	288
Incidents Management.....	289
Fetch Incidents from an Integration Instance.....	290
Classification and Mapping.....	290
Create a Search Query for Incidents.....	293
Create a Widget From an Incident.....	294
Customize Incident View Layouts.....	297
Incident Investigation.....	318
War Room Overview.....	319
Work Plan.....	325
Link Incidents.....	326
Investigate Using the Canvas.....	328
Incident Actions.....	332
Evidence Handling.....	333
Incident Tasks.....	334
Incident Fields.....	335
Incident De-Duplication.....	345
Post Processing for Incidents.....	350
Incident Access Control Configuration.....	352

Playbooks.....	355
-----------------------	------------

Playbooks Overview.....	357
Task Types.....	357
Inputs and Outputs.....	358
Field Mapping.....	358
Manage Playbook Settings.....	359
Playbook Inputs and Outputs.....	360
Playbook Tasks.....	363
Create a Conditional Task.....	363
Communication Tasks.....	365
Playbook Task Fields.....	373
Extend Context.....	379
Extend Context in a Playbook Task.....	379
Extend Context using the Command Line.....	379
Generic Polling.....	381
Prerequisites.....	381
Inputs.....	381
Generic Polling Example.....	382
Limitations of Generic Polling.....	383
Filters and Transformers.....	384
Create Filters and Transformers in a Playbook.....	384
Filter Objects Using a Transformer.....	385
Filter Objects Using the Set Command.....	387

Filter Operators.....	389
Transformers Operators.....	391
Create Custom Filters and Transformers Operators.....	395
Common Scripts to use in Automations.....	398
Work with SLAs.....	403
SLA Overview.....	405
Create an SLA Field.....	406
Manage SLA and Timer Fields in an Incident.....	408
Create an SLA Trigger.....	409
Customize SLA Scripts.....	410
Search Incidents using SLA and Timer Fields.....	411
Configure the Global Risk Threshold.....	413
Machine Learning Models.....	415
Machine Learning Models Overview.....	417
Create a Machine Learning Model.....	418
Machine Learning Model Example.....	419
Phishing Command Examples Using a Machine Learning Model.....	420
Phishing Classifier Demo.....	423
DbotPredictOutOfTheBox Parameters.....	423
DbotPredictOutOfTheBox Parameters.....	423
DbotPredictOutOfTheBox Examples.....	424
Lists.....	427
Work With Lists.....	429
Use cases.....	429
List commands.....	429
Create a List.....	430
Set the List Separator Character.....	431
Cortex XSOAR Enterprise Mobile App.....	433
Cortex XSOAR Enterprise Mobile App Overview.....	435
Android Certificate Requirements.....	435
Use the Cortex XSOAR Enterprise Mobile App.....	438
Agents.....	445
Agents Overview.....	447
Shared Agents.....	448
Configure a Shared Agent Instance.....	448
Install a Shared Agent.....	449
D2 Agent.....	451
Install a D2 Agent.....	451
Troubleshoot a Remote Installation (Windows).....	453
Agent Tools.....	454
Configure Cortex XSOAR to Use PowerShell.....	454
D2 Agent Script Commands.....	456
Return the memory dump file script.....	459
Running a Batch file Using Agent Tools.....	460
View All Running Processes Script.....	462

Logs.....	463
Audit Trail.....	465
Send the Audit Trail to an External Log Service.....	466

Cortex XSOAR Overview

Cortex XSOAR combines security orchestration, incident management, and interactive investigation into a seamless experience. The orchestration engine is designed to automate security product tasks and weave in human analyst tasks and workflows. Cortex XSOAR is powered by DBot, which learns from the real-life analyst interactions and past investigations to help SOC teams with analyst assignment suggestions, playbook enhancements, and best next steps for investigations. With Cortex XSOAR, security teams can build future proof security operations to reduce MTTR, create consistent and audited incident management process, and increase analyst productivity.

- > [Cortex XSOAR Licenses](#)
- > [Product Support Lifecycle](#)
- > [Cortex XSOAR Telemetry](#)
- > [Cortex XSOAR Concepts](#)
- > [Use Cases](#)
- > [Keyboard Shortcuts](#)
- > [How to Search in Cortex XSOAR](#)

Cortex XSOAR Licenses

- [Cortex XSOAR License Types](#)
- [Cortex XSOAR Users](#)

Cortex XSOAR License Types

The following table describes the different license types that are available for Cortex XSOAR.

	Cortex XSOAR Community Edition	Cortex XSOAR Threat Intel Mgmt.	Cortex XSOAR Starter Edition	Cortex XSOAR
Integrations	Unlimited	Unlimited	Unlimited	Unlimited
Incident Management	30 day history	30 day history	Unlimited	Unlimited
Intel Feeds	5 active feeds, each feed limited to first 100 indicators/fetch	Unlimited	5 active feeds, each feed limited to first 100 indicators/fetch	Unlimited
Automations	166 daily	Unlimited automations triggered by jobs, otherwise 166 daily	Unlimited	Unlimited
Native Intel with AutoFocus	⊘	<ul style="list-style-type: none">• Daily Threat Feed• 5k daily API points• Threat Intel Card API	⊘	<ul style="list-style-type: none">• Daily Threat Feed• 5k daily API points• Threat Intel Card API
Reporting	Incident closure only	Unlimited	Unlimited	Unlimited

- **Cortex XSOAR Community Edition:** (General free usage) Evaluates Cortex XSOAR for partner development.
- **Cortex XSOAR Threat Intel Management:** Limited to customers migrating from Minemeld and do not require case management for security orchestration, automation and response (SOAR).
- **Cortex XSOAR Starter Edition:** Relevant for customers who require case management for SOAR.
- **Cortex XSOAR:** Relevant for customers with case management for SOAR and threat intelligence needs.

Cortex XSOAR Users

Cortex XSOAR has audit users and full users.

Audit user

Audit users have read-only permission in Cortex XSOAR, meaning they do not have the ability to edit system components and data, or run commands, automations, and playbooks. Audit users can view incidents, dashboards, and reports.

Full user

Full users have read-write permission in Cortex XSOAR, meaning they have the ability to view and edit system components and data. They can investigate incidents, run automation scripts and playbooks, chat in the War Room, and so on. Full users' access to Cortex XSOAR is determined by their [assigned role](#).

Add a License

Follow these steps to add a new license to your Cortex XSOAR instance.

STEP 1 | Save the license file.

STEP 2 | Go to **Settings** > **About** > **License**.

STEP 3 | (Optional) Refresh the browser.

Product Support Lifecycle

Versioning Model

Cortex XSOAR versioning methodology is based on M.N.P:

- **M:** major version number
- **N:** minor version number
- **P:** patch-set number

Release Cycle (subject to change)

- 1 major version released annually (major version is M.x.x. For example, 2.0.0).
- 3 minor version released annually (minor version is M.N.x. For example, 2.2.0).
- Patch set version - release based on need (M.N.P. For example, 2.2.1).

Support and End-of-Life

- Major releases are supported 2 years after launch. Afterwards customers are required to upgrade to a supported major version.
- The latest and the one before latest minor release of supported major versions are supported. Customers running older minor versions are required to upgrade to the latest release of the major version.
- Only latest patch set versions are supported.

In this example, the latest version is 2.0.1. The following versions have been released:

Version	Details	Supported
2.0.1	Latest Version	Yes
2.0.0	Not the latest patch-set version of 2.0.0	No
1.2.1	Latest minor version and patch-set of 1.x.x (major version).	Yes
1.2.0	Not the latest patch-set version of 1.2.x (minor version)	No
1.1.1	Latest patch set version of one before latest minor version (1.1.x)	Yes
1.1.0	Not the latest patch-set version of the 1.1.x minor version	No
1.0.2	1.0.x minor line versions are not the latest or previous version of 1.x.x major line.	No
1.0.1		No
1.0.0	Not the latest patch-set version or minor version of 1.0.	No

General Guidelines

You should use the latest releases, as these usually include bug fixes, performance improvements, stability enhancements and may include security patches.



You can continue using a version that is out of support. However, when encountering an issue that requires Customer Support involvement, you may be asked to upgrade to a supported version before any assistance can be provided

Cortex XSOAR Telemetry

Cortex XSOAR uses telemetry to collect specific usage data. This data is analyzed and used to improve Cortex XSOAR, and to identify common usage to help drive the product roadmap.

By default, telemetry is enabled. It is recommended that you do not disable telemetry.

To disable telemetry, go to **Settings > About > Troubleshooting > Telemetry**.

Data Usage Collection

Cortex XSOAR Component	Data Collected
Playbooks	All custom playbooks, excluding encrypted playbook inputs and script arguments. The number of times each playbook was run.
Automations	All custom automation scripts in the system, excluding passwords and arguments defined as "secret".
Layouts	All custom layouts and the incident fields being used.
Classifiers	All custom mapping and classification configurations.
Integrations	Metadata for all custom integrations. The integration script is not collected.
Integration instances	Metadata for all integration instances, such as the instance name, brand, and category. Private information, such as credentials, is not collected.
Command Usage	The number of times each command is run.
Most-used commands	The command names of the most-used commands, per incident type.
Custom Fields	All custom fields, including incident fields, indicator fields, and evidence fields.
Incident Types	All custom incident types and corresponding data, such as associated playbook.
Incidents	Metadata for all incidents, including the number of incidents per incident type, the amount of time each incident stage took to resolve.
Incident Metadata	The number of incidents for each incident type, the average time of each stage.
Incident Actions	Incident creation, incident updates, whether the incident owner suggestion assignment was used, file linkage, files uploaded to the War Room.

Cortex XSOAR Component	Data Collected
Incident Cluster Usage	Modifications to the similarity filter, changes to the time frame.
Custom Indicators	All custom indicator types and corresponding data, such as type and related incidents.
Indicator Reputations	All indicator types, including name, regex, reputation command, and reputation script.
Playbooks	The number of times each playbook is run, playbook updates, playbook deletions.
Jobs	Created jobs, updated jobs.
Widgets	All custom widgets.
Dashboard	All custom dashboards.
Reports	Metadata for all scheduled reports, including name, schedule time, tags, and paper information.
Pre-Process Rules	All pre-processing rules.
Exclusion List	A summary of exclusion list rules, and exclusion count per indicator type.
Users	All user metadata. Sensitive user data is hashed, for example, user name, email address, and phone number.
Roles	All roles.
Licenses	License information.
Canvas	The total number of canvases and the number of nodes and connections for each canvas.
Version	Cortex XSOAR version and content version.
Pages	The pages of Cortex XSOAR that are accessed.
User Actions	User updates, logins, updated credentials, login method, color theme.
Settings	Update/delete: incident types, reputation (indicator types), Cortex XSOAR lists
Help Search	When the search is accessed, the search query.
Evidence	Create/update/delete evidence.
Layouts	Create/update/delete layouts.

Runtime Data Usage Collection

This data is collected every 5 minutes.

Cortex XSOAR Component	Data Collected
New Incident	Incident source, incident type, playbook name, and playbook ID.
Playbook Run	Incident source, incident type, playbook name, playbook ID, and is sub-playbook (whether it is a sub-playbook).
Command Run	Incident source, incident type, command, integration brand, trigger method (manual/automatic).
Incident Close	Incident source, incident type, open duration, and timer fields and values.
Manual Task Start	Task type, incident type, playbook name, playbook ID, and task name.
Manual Task Completion	Task type, incident type, playbook name, playbook ID, and task name.
To-Do Task	The total number of To-Do tasks. Whether the DBot suggested was selected.

Cortex XSOAR Concepts

It is important to familiarize yourself with Cortex XSOAR components, UI terminology, and key concepts.

Incidents

Potential security data threat that SOC administrators identify and remediate. There are several incident triggers, including:

- SIEM alerts
- Mail alerts
- Security alerts from third-party services, such as SIEM, mail boxes, data in CSV format, or from the Cortex XSOAR RESTful API

Cortex XSOAR includes several out-of-the-box incident types, and users can add custom incident types with custom fields, as necessary.

Incident Fields

Incident Fields are used for accepting or populating incident data coming from incidents. You create fields for information you know will be coming from 3rd party integrations and in which you want to insert the information.

Incident Lifecycle

Cortex XSOAR is an orchestration and automation system used to bring all of the various pieces of your security apparatus together. Using Cortex XSOAR, you can define integrations with your 3rd-party security and incident management vendors. You can then trigger events from these integrations that become incidents in Cortex XSOAR. Once the incidents are created, you can run playbooks on these incidents to enrich them with information from other products in your system, which helps you complete the picture. In most cases, you can use rules and automation to determine if an incident requires further investigation or can be closed based on the findings. This enables your analysts to focus on the minority of incidents that require further investigation.

Integrations

Third-party tools and services that the Cortex XSOAR platform orchestrates and automates SOC operations. In addition to third-party tools, you can create your own integration using the Bring Your Own Integration (BYOI) feature.

The following lists some of the integration categories available in Cortex XSOAR. The list is not exhaustive, and highlights the main categories:

- Analytics and SIEM
- Authentication
- Case Management
- Data Enrichment
- Threat Intelligence
- Database
- Endpoint
- Forensics and Malware Analysis
- IT Services
- Messaging

-
- Network Security
 - Vulnerability Management

Integration Instance

A configuration of an integration. You can have multiple instances of an integration, for example, to connect to different environments. Additionally, if you are an MSSP and have multiple tenants, you could configure a separate instance for each tenant.

Playbooks

Cortex XSOAR Playbooks are self-contained, fully documented prescriptive procedures that query, analyze, and take action based on the gathered results. Playbooks enable you to organize and document security monitoring, orchestration, and response activities. There are several out-of-the-box playbooks that cover common investigation scenarios. You can use these playbooks as-is, or customize them according to your requirements. Playbooks are written in YAML file format using the COPS standard.

A key feature of Playbooks is the ability to structure and automate security responses, which were previously handled manually. You can reuse Playbook tasks as building blocks for new playbooks, saving you time and streamlining knowledge retention.

Automations

The Automation section is where you manage, create, and modify scripts. These scripts perform a specific action, and are comprised of commands associated with an integration. You write scripts in either Python or JavaScript. Scripts are used as part of tasks, which are used in playbooks and commands in the War Room.

Scripts can access all Cortex XSOAR APIs, including access to incidents, investigations, share data to the War Room, and so on. Scripts can receive and access arguments, and you can password protect scripts.

The Automation section includes a Script Helper, which provides a list of available commands and scripts, ordered alphabetically.

Commands

Cortex XSOAR has two different kinds of commands:

- system commands - Commands that enable you to perform Cortex XSOAR operations, such as clearing the playground or closing an incident. These commands are not specific to an integration. System commands are entered in the command line using a `/`.
- external commands - Integration-specific commands that enable you to perform actions specific to an integration. For example, you can quickly check the reputation of an ip. External commands are entered in the command line using a `!`. For example, `!ip`.

War Room

The War Room is a collection of all investigation actions, artifacts, and collaboration pieces for an incident. It is a chronological journal of the incident investigation. You can run commands and playbooks from the War Room and filter the entries for easier viewing.

Indicators and Indicator Types

DBot can simplify your incident investigation process by collecting and analyzing information and artifacts found in War Room entries. Cortex XSOAR analyzes indicators to determine whether they are malicious. Using indicator types reveals predefined, regular expressions in the War Room.

Hits are indicators that are determined to have a bad reputation, and were previously identified in the network. The reputation is the indicator's level of maliciousness, determined manually or by hypersearch scripts. If a hypersearch script identifies an indicator, the source is DBot.

There are many out-of-the-box indicator types, but you can add custom indicator types as necessary. The following is a list of some of the indicator types, but the list is not exhaustive:

- IP address (IP4, IP6)
- Registry path
- URL
- Email
- File hash (SHA-1, MD5)
- Domains
- CIDR

When you add an indicator type, you can add enhancement and reputation scripts. Enhancement scripts enable you to gather additional data about the highlighted entry in the War Room. Reputation scripts calculate the reputation score for an entry that DBot analyzed, for example, DatalPReputation, which calculates the reputation of an IP address.

Playground

The playground is a non-production environment where you can safely develop and test automation scripts, APIs, commands, and more. It is an investigation area that is not connected to a live (active) investigation.

To erase a playground and create a new one, in the Cortex XSOAR CLI run the `/playground_create` command.

Jobs

You can create scheduled events in Cortex XSOAR using jobs. Jobs are triggered either by time-triggered events or feed-triggered events. For example, you can define a job to trigger a playbook when a specified TIM feed finishes a fetch operation that included a modification to the list.

Use Cases

This section includes common Use Cases for the different categories of Cortex XSOAR integrations. While this list is not meant to be exhaustive, it's a good starting point for you to understand what use cases could be supported by your integration.

Analytics and SIEM

Top Use Cases:

- Fetch Incidents with relevant filters
- Create, close and delete incidents/events/cases
- Update Incidents - Update status, assignees, Severity, SLA, etc.
- Get events related to an incident/case for enrichment/investigation purposes
- Query SIEM (consider aggregating logs)

Please Note: This will normally include the Fetch Incidents possibility for the instance. It can also include list-incidents or get-incident as integration commands. Important information for an Event/Incident.

Analytics & SIEM Integration Example: ArcSight ESM

Authentication

Top Use Cases:

- Use credentials from authentication vault in order to configure instances in Cortex XSOAR (Save credentials in: Settings -> Integrations -> Credentials) The integration should include the isFetchCredentials Parameter, and other integrations that will use credentials from the vault, should have the 'Switch to credentials' option
- Lock/Delete Account - Give option to lock account (credentials), and unlock/undelete
- Reset Account - Perform a reset password command for an account
- List credential names - Do not post the actual credentials. (For example - Credential name: McAfee ePO, do not show actual username and password.)
- Lock Vault - In case of an emergency (if the vault has been compromised), allow the option to lock + unlock the whole vault
- Step-Up authentication - Enforce Multi Factor Authentication for an account

Authentication Integration Example: CyberArk AIM

Case Management

Top Use Cases:

- Create, get, edit, close a ticket/issue, add + view comments
- Assign a ticket/issue to a specified user
- List all tickets, filter by name, date, assignee
- Get details about a managed object, update, create, delete
- Add and manage users

Case Management/Ticketing Integration Example: ServiceNow

Data Enrichment & Threat Intelligence

Top Use Cases:

-
- Enriching information about different IOC types: Upload object for scan and get the scan results. (If there's a possibility to upload private/public, default should be set to private). Search for former scan results about an object (This way you can get information about a sample without uploading it yourself). Enrich information and scoring for the object.
 - Add/Search for indicators in the system
 - Add indicators to the exclusion list
 - Calculate DBot Score for indicators

Data Enrichment & Threat Intelligence Integration Example: VirusTotal

Email Gateway

Top Use Cases:

- Get message – Download the email itself, retrieve metadata, body
- Download attachments for a given message
- Manage senders – Block/ Allow specified mail senders
- Manage URLs – Block/ Allow the sending of specified URLs
- Encode/ Decode URLs in messages
- Release a held message (The gateway can place suspicious messages on hold, and sometimes they would need to be released to the receiver)

Email Gateway Integration Example: MimeCast

Endpoint

Top Use Cases:

- Fetch Incidents & Events
- Get event details (from specified incident)
- Quarantine File
- Isolate and contain endpoints
- Update Indicators (Network, hashes, etc.) by policy (can be block, monitor) – deny list
- Add indicators to the exclusion list
- Search for indicators in the system (Seen indicators and related incidents/events)
- Download file (based on hash, path)
- Trigger scans on specified hosts
- Update .DAT files for signatures and compare existing .DAT file to the newest one on the server
- Get information for a specified host (OS, users, addresses, hostname)
- Get policy information and assign policies to endpoints

Endpoint Integration Examples: Cortex XDR, Tanium and Carbon Black Protection

Forensics and Malware Analysis

Top Use Cases:

- Submit a file and get a report (detonation)
- Submit a URL and get a report (detonation)
- Search for past analysis (input being a hash/URL)
- Retrieve a PCAP file
- Retrieve screenshots taken during analysis

Sandbox Integration Example: Cuckoo Sandbox

Network Security (Firewall)

Top Use Cases:

- Create block/accept policies (Source, Destination, Port), for IP addresses and domains
- Add addresses and ports (services) to predefined groups, create groups, etc.
- Support custom URL categories
- Fetch network logs for a specific address for a configurable time frame
- URL filtering categorization change request
- Built in blocked rule command for fast-blocking
- If there is a Management FW, allow the option to manage policy rules through it

Network Security Firewall Integration Example: Palo Alto Networks PAN-OS

Network Security (IDS/IPS)

Top Use Cases:

- Get/Fetch alerts
- Get PCAP file, packet
- Get network logs filtered by time range, ip addresses, ports, etc.
- Create/manage/delete policies and rules
- Update signatures from an online source / upload + Get last signature update information
- Install policy (if existing)

Network Security (IPS/IDS) Integration Example: Protectwise

Vulnerability Management

Top Use Cases:

- Enrich asset – get vulnerability information for an asset (or a group of assets) in the organization
- Generate/Trigger a scan on specified assets
- Get a scan report including vulnerability information for a specified scan and export it
- Get details for a specified vulnerability
- Scan assets for a specific vulnerability

Vulnerability Management Integration Example: Tenable.io

Keyboard Shortcuts

There are several keyboard shortcuts that enable you to quickly navigate and manage Cortex XSOAR.

Playbooks, Scripts, and CLI

Description	Mac	Windows	Available in Pages
Fast Navigation	Command-K	Ctrl + K	All
Focus on the CLI	Command-;	Ctrl + ;	All
Save the Playbook	Command-S	Ctrl + S	Playbooks
Auto-Align the Playbook	Command-L	Ctrl + L	Playbooks (edit mode)
Save the Automation Script	Command-S	Ctrl + S	Automation
Create a New Automation script	Command-I	Ctrl + I	Automation
Create a New Incident	Command-I	Ctrl + I	Home, Incidents
Open the Markdown Box and the Toolbar in CLI	Ctrl-M	Ctrl + M	CLI area in all pages
Toggle Full View/Right Side Menu View	Option-F	Alt + F	Dashboard, Incident Page, Automations, Playbooks

Switch Between Incident Pages

Description	Mac	Windows
Navigate to the Incident Details Page	Option-1	Alt + 1
Navigate to the War Room Page	Option-2	Alt + 2
Navigate to the Workplan Page	Option-3	Alt + 3
Navigate to the Evidence Board Page	Option-4	Alt + 4
Navigate to the Related Incidents Page	Option-5	Alt + 5

Description	Mac	Windows
Navigate to Canvas Page	Option-6	Alt + 6

Open and Close Right Shoulders

Description	Mac	Windows
Open/Close the Incident Summary Quick View	Option-Q	Alt + Q
Open/Close the Systems Windows	Option-W	Alt + W
Open/Close the Incident Team Window	Option-E	Alt + E
Open/Close the Incident Context View	Option-R	Alt + R

How to Search in Cortex XSOAR

Cortex XSOAR comes with a very powerful search capability. You can search for data in Cortex XSOAR in the following ways:

- Using the search query: searches for information using the [Lucene query syntax](#). The search query appears in the Indicators, Incidents, Jobs, Playbooks, Automation, and the Evidence Board pages. For example, to search for all incidents that have the status as pending and are critical, type **status:Pending and severity:Critical**.

ID ↓	Name	Type	Severity	Status
#1656	Sample Incident - Phishing	Phishing	Critical	Pending
#1649	Sample Incident - Phishing	Phishing	Critical	Pending

- Using the search box: searches for incidents, entries, evidence, investigations, and indicators in Cortex XSOAR. The search box appears in the top right hand corner in every page. You can either type free text or search using the search query format (use the arrow keys to assist you in the search). For example, **incident.severity:Low** searches for all incidents that have **low** in the severity category.

Search: displaying 50 out of 452 results. Size: 50 × Last 30 days ▾

ID: #6
Name: Sample Incident - Phishing
Incident Investigate

ID: #11
Name: Sample Incident - Malware
Incident Investigate

- Using a general search. For example, when searching for a table in the **Users** tab, searching for a widget, or a task in a playbook, etc.

Using the Search Query

The search follows the [Lucene query syntax](#). The search is performed on certain pages such as incidents, indicators, etc, or the entire data (titles, entries, chats, etc.).

Basic syntax of the search

You can add some of the following inputs, when searching for data:

Input	Description
Add text	Type any text. The results show all data where one of the words appears. For example, the search low virus returns all data where either the string, low or the string, virus appears.
and	Searches for data where all conditions are met. For example, status:Active and severity:High finds all incidents with an active status that has an high severity.
or	Searches for data where either conditions are met. For example, status:Pending and severity:High or severity:Critical finds for all incidents with a pending status and with severity high or critical.
* ?	Wildcard search: * and ? should be used when searching for partial strings. For example, when searching for all scripts that start with AD, use AD* . If you need to search for script, which contains "get", search for *get* .
""	An empty value.
-	Excludes from any search. For example in the Incidents page the -status:closed -category:job searches for all incidents that are not closed and for categories other than jobs.
Relative time. For example: <ul style="list-style-type: none">• "half an hour ago"• "1 hour ago"• "5 minutes ago"• "10 days ago"• "five days ago"• "5 seconds ago"• "two weeks ago"• "a month ago"• "a few months ago"• "one year ago"• a week ago	Relative time in natural language can be used in search queries. Time filters - < and > can be used when referring to a specified time, such as dueDate:>="2018-03-05T00:00:00 +0200" .  <i>When adding some fields, such as Occurred you can enter the date from the calendar. You can also filter the date when the results are displayed.</i>

Configure System Notifications

If there are multiple instances of a mail sender in Cortex XSOAR, you can choose which email sender should send the notification by configuring the `server.notification.using.send-mail` key in the advanced server configuration settings.

If you do not configure the advanced server setting, Cortex XSOAR uses the first email integration it finds to send the system notifications.

STEP 1 | Select **Settings** > **About** > **Troubleshooting** > **Add Server Configuration**.

STEP 2 | Enter the following key and value:

Key	Value
<code>server.notification.using.send-mail</code>	The mail sender instance name.

Install DBot for Slack

The following steps explain how to install and configure the DBot app for Slack.

By default, DBot monitors all channels in non-verbose mode.

STEP 1 | Go to <https://dbot.demisto.com> to install the DBot app for Slack.

STEP 2 | Click the **Add to Slack** button.

STEP 3 | When prompted, review the required permissions and click **Allow**.

STEP 4 | Go to <https://dbot.demisto.com/conf> to configure the DBot app for Slack.

Select which channels and conversations to monitor.

Single Server Deployment

- > System Requirements
- > Install Cortex XSOAR for a Single Server Deployment
- > Install Cortex XSOAR Offline
- > Post-Installation Checklist
- > AWS EC2 Deployment Guidelines
- > Upgrade the Cortex XSOAR Server
- > Uninstall Cortex XSOAR

System Requirements

Cortex XSOAR requires the following software and hardware. Ensure you meet all minimum system requirements.

- [Cortex XSOAR Server](#)
- [Cortex XSOAR Engine](#)
- [Web Browsers](#)
- [Required URLs](#)

Cortex XSOAR Server

Cortex XSOAR server has specific operating system and hardware requirements.

Operating Systems

You can deploy Cortex XSOAR on the following operating systems and must meet the minimum hardware requirements:

Operating System	Supported Versions
CentOS	7.x and 8
Ubuntu	16.04 and 18.04
RHEL	7.x and 8
Oracle Linux	7.x
Amazon Linux	2

Hardware Requirements

Component	Dev Environment Minimum	Recommended
CPU	8 CPU cores	16 CPU cores
Memory	16GB RAM	32GB RAM
Storage	500GB SSD	1TB SSD with minimum 3k dedicated IOPS

Cortex XSOAR Engine

Cortex XSOAR engines are compatible with the following operating systems and must meet the minimum hardware requirements.

- Windows
- MacOS
- Linux

Hardware Requirements

Component	Dev Environment Minimum	Production Minimum
CPU	8 CPU cores	16 CPU cores
Memory	16GB RAM	32GB RAM
Storage	20GB	20GB

Web Browsers

Cortex XSOAR supports the following web browsers:

Web Browser	Version
Chrome	46.x and later
Safari	6.x and later
Firefox	43.x and later
Microsoft Edge	Latest version
Internet Explorer	11

Required URLs

You need to allow the following URLs for Cortex XSOAR to operate properly.

Function	Service/Port	Direction
Web interface	HTTPS (443/configurable)	Users > Server
Engine connectivity	HTTPS (443/configurable)	Engine > Server
Integrations	Integration-specific ports	Server > Integration
REST API	HTTPS (443/configurable) https://api.demisto.com/	Third Party > Server
Docker	<ul style="list-style-type: none"> https://registry-1.docker.io https://auth.docker.io (This URL may change according to Docker's discretion). https://production.cloudflare.docker.com (This URL may change according to Docker's discretion). 	Server > Docker URLs
Cortex XSOAR content updates	<ul style="list-style-type: none"> https://api.github.com 	Server > GitHub

Function	Service/Port	Direction
	<ul style="list-style-type: none"> • https://github-production-release-asset-2e65be.s3.amazonaws.com/ 	
Marketplace	<ul style="list-style-type: none"> • xsoar.pan.dev/marketplace-index-mapping (download packs and view marketplace) • storage.googleapis.com (download packs and view marketplace) • xsoar-authentication-proxy.paloaltonetworks.com (login and register users) • xsoar-marketplace-review.paloaltonetworks.com (review packs) • xsoar-marketplace-subscriptions.paloaltonetworks.com (subscribe to packs) • xsoar-premium-content-gateway.paloaltonetworks.com (download premium packs, including free premium packs) • xsoar-contribution-dev.web.app (contribute packs) 	Server > Marketplace

Performance Benchmark

Cortex XSOAR is designed to maximize performance and enable scalability, delivering factors to provide the best experience and performance, and such benchmarking process is conducted annually to ensure the best performance levels.

Cortex XSOAR performance is determined by compute, memory, and HD performance. Each component can impact a different part of the system, therefore it is important to ensure that you deploy Cortex XSOAR on an infrastructure that [meets all requirements](#).

The amount of data each incident holds can have a significant impact on performance and disk space of the system. To achieve optimal performance and disk usage, we recommend that an incident not be larger than 0.5mb.

Disk Usage

The required disk space for each incident varies based on the number of integrations and the size and complexity of the playbook. We simulated the number of incidents and their respective size in the disk. For the simulation we used out-of-the-box integrations and an example phishing playbook (see Simulation Incidents and Required Disk Size table).

The incidents were generated using genuine phishing emails of various sizes, which averaged 4KB.

The below values show the disk space for incidents after ingestion and playbook run, without Demisto data compression. A plain incident before ingestion and playbook run averages 3KB in the file system, and depends on the data received from the SIEM.



You should not compare disk usage tests between versions. Different tests are performed for each version.

Product Version	10,000	40,000	100,000
5.0	22 GB	90 GB	225 GB
5.5	27 GB	108 GB	270 GB

Performance Benchmark Test

The benchmark test was performed on dedicated virtual servers running Amazon Linux 2. The following tests were performed on dedicated machines with recommended specifications. The tests were performed with a dedicated SSD hardware IOPS granting persistence input/output operations per second, rather than possible varying IOPS rate, which is common in cloud machines.

The virtual server specifications were:

- 16 CPU cores
- 32 GB RAM
- 1 TB SSD (GP2)
- Cortex XSOAR v5.5

Benchmark Process

The benchmarking process is executed using an automated test, which allows for batch incident mapping and classifying, ingestion, and execution of a specified playbook. We then measure each step in the incident lifecycle and measure the total time it took from the first incident received to the last incident closed.

The same three tests were performed on two different environments, a single-server environment and a multi-repo environment. Each environment contained only out-of-the-box integrations, scripts, and commands.

- Ingest and run 50 incidents with an automated playbook to completion.
- Ingest and run 100 incidents with an automated playbook to completion.
- Ingest and run 500 incidents with an automated playbook to completion.

Ingestion

Incidents are ingested via HTTP REST request.

Process

Each incident automatically triggers the default phishing playbook (complicated level), which performs the actions listed below. These actions are performed simultaneously for all incidents, on each test.

- Parse and process the email
- Auto-run IOC extraction and reputation checks for all indicators
- Extract attachments
- Calculate incident severity based on IOCs
- Notify users (administrators and the email sender) about the progress of the incident
- Close the incident

Benchmark Results

The results are the average time for each test.

The numbers specified were processed without data compression. The results might vary based on the machines' hardware specifications, system configurations, Docker version, and the type of actions performed.

Cortex XSOAR utilizes free memory and available resources to enable faster system performance, including cache, containers manipulation, and more.

Single-server Tests

Number of incidents executed in parallel	Time to complete ingestion	Time to complete playbook
50	814ms	2m49s
100	1.84s	4m43s
500	10.15s	31m22s

Distributed Database Tests

The environment that was tested consisted of four servers: 1 app server, 1 main DB node, 2 DB nodes)

- 1 application server
- 1 main DB node

-
- 2 DB nodes

Number of incidents executed in parallel	Time to complete ingestion	Time to complete playbook
50	0.5s	2m12s
100	1.05s	4m42s
500	5.2s	22m59s

Install Cortex XSOAR for a Single Server Deployment

In a standard Cortex XSOAR deployment, the app server and database server are installed on the same machine.

Installation File Structure

This is the file and folder structure in a standard Cortex XSOAR installation.

Asset	Path
Binaries	<code>/usr/local/demisto</code>
Data	<code>/var/lib/demisto</code>
Logs	<code>/var/log/demisto</code>
Configuration	<code>/etc/demisto.conf</code> (will not be created if defaults are selected during installation)
Reports	<code>/tmp/demisto_install.log</code>
Install Log	<code>/tmp/demisto_install.log</code>

Prerequisites

Verify the following information and requirements before you install Cortex XSOAR.

- Your deployment meets the [minimum system requirements](#).
- You have root access.
- The production server has Python 2.7 or 3.x.
- (CentOS 8) Open the installation script by running the `sudo yum install tar` command.

STEP 1 | Download the server package from the link that you received from Cortex XSOAR Support.

```
demistoserver.xxxx.sh
```

STEP 2 | As root user, execute the `.sh` file.

```
./demistoserver-xxxx.sh
```

STEP 3 | Accept the EULA and complete the installation process.

STEP 4 | Run the `chmod +x demistoserver-xxxx.sh` command to convert the `.sh` file to an executable file.

STEP 5 | In a web browser, go to the `https:// serverURL: port` to verify that Cortex XSOAR was successfully installed.

Installer Flags

Flags that follow the -- separator

Flag	Type	Description
-C	N/A	(CentOS) Tells yum to run entirely from system cache, and does not download or update any headers unless it has to perform the requested action.
-backup	Boolean	Whether to back up server data when upgrading the Cortex XSOAR server.
-backup-tenants	Boolean	Whether to back up server data for tenants when upgrading the Cortex XSOAR server. Default is true.
-conf file	String	The server .conf file. The default location is /etc/demisto.conf.
-db-address1	String	The host name or IP address of the remote database.
-db-any-certificate	Boolean	Whether to trust any certificate when communicating with the database. Default is true.
-db-conn-port	String	The secure connection port for the remote database. Default is 50001.
-db-only	Boolean	Whether to set up only the Cortex XSOAR database server. Applicable for remote database deployment. Default is false.
-db-port	String	The port of the remote database. Default is 443.
-db-secret	String	The secret for the remote database.
-db-use-proxy	N/A	The proxy is used when communicating with the database.

Flag	Type	Description
-demisto-no-gpg-check	N/A	Tells yum to disable the gpg signature when checking for the Cortex XSOAR RPM.
-distro	String	Forces the Linux distro (such as CentOS, Debian, or Debian New).
-do-not-start-server	N/A	Prevents starting the server when the installation or upgrade is complete.
-dr	N/A	Sets up the disaster recovery server.
-elasticsearch-url	String	Elasticsearch URL addresses (comma-separated). For example, <code>http://test1:9200,http://test2:9200</code>
-elasticsearch-api-key	String	The Elasticsearch API key, which should be used in licensed versions.
-elasticsearch-username	String	The Elasticsearch username.
-elasticsearch-password	String	The Elasticsearch password.
-elasticsearch-proxy	Boolean	Whether to use a proxy when communicating with Elasticsearch. Can be true or false . Default is false .
-elasticsearch-insecure	Boolean	Whether to trust any certificate when communicating with Elasticsearch. Can be true or false . Default is true .
-elasticsearch-	Integer	The amount of time (in seconds) before Elasticsearch times out. Default is 20 seconds.
-external-address	String	The external address of the instance during installation.
-h	N/A	Displays installation help.
-index-entries	Boolean	Whether to index entries.
-otc	String	The one-time configuration file. The default location

Flag	Type	Description
		<code>is /var/lib/demisto/otc.conf.json.</code>
<code>-prev-uninstall-script</code>	String	The path for the uninstall script. The default location is <code>/var/lib/dpkg/info/demistoserver.postrm.</code>
<code>-purge</code>	N/A	Removes the existing Cortex XSOAR installation.
<code>-restore-entries</code>	Boolean	Restores the entries index. If false, it prevents restoring the entries index. The default is true.
<code>-server-only</code>	N/A	(For remote database deployments) Sets up only the Cortex XSOAR server.
<code>-tools</code>	Boolean	Installs the required tools. The default is true.
<code>-use-prev-uninstall-script</code>	N/A	(For Cortex XSOAR upgrades) The script that deletes the Cortex XSOAR user and group is not run.
<code>-y</code>	N/A	Answer all installer questions with y/yes, including the Cortex XSOAR EULA.

Flags that precede and include the `--` separator

Use the following flags to get help or information about the `./demistoserver-5.5-XXXXX.sh` file.

Flag	Description
<code>--help</code>	Prints a message.
<code>--info</code>	Prints embedded information, including title, default target directory, or embedded script.
<code>--lsm</code>	Prints an embedded LSM entry, if one exists.
<code>--list</code>	Prints a list of files located in the archive.
<code>--check</code>	Checks the integrity of the archive.

Use the following flags to run the `./demistoserver-5.5-XXXXX.sh` file.

Flag	Description
<code>--confirm</code>	Prompts you to confirm before running an embedded script.
<code>--quiet</code>	Prints only error messages.
<code>--accept</code>	Accepts the Cortex XSOAR license.
<code>--noexec</code>	Embedded scripts are not run.
<code>--keep</code>	The target directory is not deleted after running an embedded script.
<code>--noprogess</code>	Progress is hidden during decompression.
<code>--nox11</code>	An xterm is not spawned.
<code>--nochown</code>	Extracted files are not given to users.
<code>--nodiskspace</code>	Disk space is not checked for available space.
<code>--target dir</code>	Extracts directly to an absolute or relative target directory.
<code>--tar arg1[arg2...]</code>	Accesses the archive contents.

Install Cortex XSOAR Offline

These instructions apply when using the Cortex XSOAR installer script for installation with no internet connectivity. The dependencies might change across releases. Always verify that your yum repo packages are updated according to the [dependency list](#).

To install Cortex XSOAR without an internet connection, you need to download CentOS packages and Docker dependencies before you install Cortex XSOAR server.

Docker is used to run Python scripts in an isolated container.

STEP 1 | Create a local yum repo with the [required dependencies](#) for your deployment type.

STEP 2 | Download and install the Docker dependencies.

1. In the Cortex XSOAR installer download link, append the link with `downloadName=dockerimages`.
2. Run the `docker load -i <YOUR.DOCKER.FILE>.tar` command from the Cortex XSOAR server machine.
3. Verify that the local yum repository is properly configured before you run the Cortex XSOAR installer.

STEP 3 | Run the command for your deployment to install Cortex XSOAR.

Deployment Type	Command
CentOS	<code>./demistoserver-x.x-xxxxx.sh -- -tools=false -C</code>
Red Hat	<code>./demistoserver-x.x-xxxxx.sh -- -tools=false</code>

STEP 4 | Run the `repoquery -a --installed` command to verify that the required dependencies were successfully installed.

Dependencies for Offline Installation

RPM dependencies

The following dependencies are required for Red Hat and CentOS deployments.

- systemd
- samba-client
- samba-common-tools
- samba
- xmlsec1
- xmlsec1-openssl
- rpm-build
- libcap
- file-devel
- fontconfig
- expat
- libpng

-
- freetype
 - java-1.8.0-openjdk
 - git

Debian dependencies

The following dependencies are required for Debian and Ubuntu deployments.

- systemd-services
- smbclient
- xmlsec1
- rpm
- libcap2-bin
- file
- libfontconfig1
- libexpat1
- libpng12-0
- libfreetype6
- openjdk-8-jre
- git

Post-Installation Checklist

- Server Post-Installation Health Check
- Monitor Cortex XSOAR Components
- [HTTPS with a Signed Certificate](#)
- Configure Websockets

Server Post-Installation Health Check

After you install the Cortex XSOAR server, it is recommended that you verify the installation by checking several systems and running several commands.

If you experience issues with any of these health checks, contact the Cortex XSOAR support team.

Check the Docker sub-system

Run these commands in the playground.

- `/docker_images`: Verify that either a list of Docker images or an empty list is returned.
- `!py script="demisto.results('hello world')"`: Verify that `hello world` is returned, not an error.
- `sudo docker info`: Check for warnings or errors.

If no Docker images were installed, then the `!py script="demisto.results('hello world')"` command might take several minutes to execute, and might fail the first time it is run. If the command fails, rerun the command after five minutes.

Verify Integration Tests

Create an instance of each of the following integrations and test each of these integrations by clicking the **Test** button in the integration instance.

- urlscan.io
- ipinfo
- PhishTank
- OpenPhish
- Rasterize

Also, run `!FailedInstances` in the playground to test all configured integrations and check outputs to see that there are no errors returned.

Run Commands in the Playground

We recommend that you run the following commands in the playground.

Command Name	Related Integration	Full Command	Description
url	urlscan.io	<code>!url url="https://google.com"</code>	Submits a URL to scan.
	PhishTank		Checks URL reputation
	OpenPhish		

Command Name	Related Integration	Full Command	Description
ip	ipinfo	<code>!ip ip="8.8.8.8"</code>	Checks the IP reputation (when information is available, returns a JSON with details). Uses all configured Threat Intelligence feeds.
rasterize	Rasterize	<code>!rasterize url="https://google.com"</code> <code>!rasterize-email htmlBody="<h1>hello world</h1>"</code>	Converts the contents of a URL to an image file or a PDF file. Converts the body of an email to an image file or a PDF file.
		<code>!Ping address=8.8.8.8</code>	Verifies accessibility of the host from the Internet .

Check the reporting sub-system

Run any report and verify that the PDF output resolves correctly.

Content

- Verify that your content is up to date. Navigate to the **Playbooks** section and click **Check for new content**.
- Verify that you see automation scripts in the **Automations** section.
- Verify that you see playbooks in the **Playbooks** section.
- Verify that you see dashboard widgets in the **My Dashboards** section.

Monitor Cortex XSOAR Components

Cortex XSOAR provides several tools to monitor your system's health. For self-hosted instances, use your organization's standard system health monitoring tools to monitor the performance of your server's disk, CPU, and RAM.

Verify That The Server Is Up and Running

To check that the server is up and running, navigate to the following URL: `https://<your.demisto.hostname>/health`. If the server is up and running, the system returns HTTP status code 200 OK.

Integration Fetch Error Notifications

You can specify a comma-separated [list of users to notify](#) when an integration fails to fetch events.

Monitor Engine Hosts

For all instances, in addition to monitoring your Cortex XSOAR server, it is important to monitor the health of your engine hosts. You should ensure that the service is up and running, and that the disk, CPU, and RAM are not being over-utilized. For additional ways to monitor and manage your engine hosts, see the following sections in *Managing Cortex XSOAR Engines*:

- Get Engine Logs

-
- Notify Users When an Engine Disconnects

Post-Installation Health Checks

After you install Cortex XSOAR, we recommend that you run several health checks, for content, integrations, Docker images, and so on, to verify that your environment is working as expected. For more information, see [Server Post-Installation Health Check](#).

If you experience issues with any of these health checks, contact the Cortex XSOAR support team.

HTTPS with a Signed Certificate

By default the server uses a self-signed certificate for a secure HTTP connection. Only TLS 1.2 is supported.

If you want to use your own server certificate (X.509 certificates), it is recommended to replace `~/usr/local/demisto/cert.key` with the private key and `~/usr/local/demisto/cert.pem` with the certificate. To create your own certificate and key, see [Create a Private Key and Certificate Signing Request \(CSR\)](#).

For the certificate PEM file, you must concatenate the certificate chain one after the other in the file. The SSL certificate should come first, and the CA certificate(s) second. Only the certificate itself is needed, i.e. the text between and including `"-----BEGIN CERTIFICATE-----"` and `"-----END CERTIFICATE-----"`.

cert.pem



You can store the key and certificate in a different location, by changing the `/etc/demisto.conf` file and adding the locations below:

```
{  
  "Security": {  
    "CertFile": "",  
    "KeyFile": ""  
  }  
}
```

Ensure both files have the correct ownership: `demisto:demisto`

If your private key is encrypted, you need to add the key password to the one-time-configuration (OTC) file located in `/var/lib/demisto/otc.conf.json`. After the file is saved and the Cortex XSOAR server is restarted, the OTC file is automatically deleted. Add the following content to the OTC file.

```
{"keypass": "certpassword"}
```



Cortex XSOAR server does not support PKCS#8 encrypted PEM files. To validate that the file is supported, check that the "DEK-Info" header exists.



When using a Safari browser, the self-signed certificate must be added to the OS Keychain.

Create a Private Key and Certificate Signing Request (CSR)

Follow these instructions to create a private key and certificate signing request.

STEP 1 | On a SSH session to the Cortex XSOAR server, generate the private certificate by running the following command.

```
openssl genrsa -out DemistoPrivateKey.key 2048
```

The RSA private key is generated.

STEP 2 | Generate the Certificate Signing Request (CSR) by running the following command.

```
openssl req -new -sha256 -key DemistoPrivateKey.key -out  
DemistoPrivateCert.csr
```

STEP 3 | Follow the on-screen instructions.

The CSR is sent to the certificate signing authority. The CA authority sends the certificate by email in different formats. Use the certificate in X.509 format with the .pem extension.

STEP 4 | Replace the existing internal certificate in `/usr/local/demisto/cert.pem` and key in `/usr/local/demisto/cert.key` with the newly generated private certificate and key.

STEP 5 | Restart the Cortex XSOAR server.

Keep the certificate and key in a place other than `/usr/local/demisto`.

Troubleshoot Creating a Private Key and CSR

After the newly generated certificate key pair is copied to `/usr/local/demisto`, if the browser does not show the new certificate, do one or more of the following:

- Check whether the FQDN specified in the certificate is the same as the FQDN of the Cortex XSOAR server.
- Check whether there are any other certificates or keys in `/usr/local/demisto`, other than the ones generated recently for the Cortex XSOAR server. If so, remove or move them to another folder on the server.
- On your browser on which you are trying to load Cortex XSOAR, clear cookies and other data. For example, in Chrome, go to **Settings > Advanced > Clear Browsing data > Clear data**.
- If the Cortex XSOAR server is behind a load balancer, re-upload the certificate on the load balancer. For example, if the Cortex XSOAR server is behind the ELB (Elastic Load Balancing), re-import the certificate on ELB (Elastic Load Balancing) on the Amazon Certificate Manager AWS console.

AWS EC2 Deployment Guidelines

AWS EC2 deployments have specific technical and sizing requirements, and a set of Cortex XSOAR best practices.

After you configure and verify your AWS EC2 deployment, install the Cortex XSOAR Server.

Supported AWS EC2 Instance Types

Cortex XSOAR server installation is supported on the following Amazon EC2 instance types.

Instance Type	Instance Name
M5	m5.4xlarge
	m5.12xlarge
	m5.24xlarge
M4	m4.4xlarge
	m4.10xlarge
	m4.16xlarge
C5	c5.4xlarge
	c5.9xlarge
	c5.18xlarge
C4	c4.4xlarge
	c4.8xlarge
R4	r4.4xlarge
	r4.8xlarge
	r4.16xlarge
T2	Only use this instance type for evaluation and testing.

Supported EC2 Storage

We recommend that you use EBS-optimized2 instances (included in C4,M4 instance family types).



Minimum disk space for production environment is 500 GB.

Network and Security

Inbound connections: Cortex XSOAR server requires that you open HTTPS port 443 for inbound connections.

Outbound connections: outbound ports depending on your integrations and specific requirements. By default, AWS opens all traffic.

Supported OS

Cortex XSOAR supports server installation on these Linux distributions.

- CentOS v7.2 and later (recommended)
- Ubuntu Server v14.04 LTS and later (recommended)
- RHEL v7.2 and later (Docker CE is not supported. You need Docker EE to run specific Docker-dependent integrations and automations. For more information, contact the Demisto support team).

AWS Backup

You can use Amazon EBS snapshots³ to back up your AWS EC2 instances. Use snapshots for incremental backups, point-in-time restoration, and to view history data.

Upgrade the Cortex XSOAR Server

The installer automatically detects the existing configurations and applies them to the upgraded server.

STEP 1 | Back up the entire system.

STEP 2 | Download and run the installer file.

```
sudo ./demistosever-5.5-XXXX -- -y
```

Uninstall Cortex XSOAR

When you uninstall Cortex XSOAR, configuration files and files that were created by engines (i.e., log files) are not removed. If you uninstall and then re-install Cortex XSOAR, make sure the `d1.conf` and other related files are removed.

The uninstall command supports the `y` flag.

Run the following command to uninstall Cortex XSOAR.

```
sudo ./demistoserver-xxxx.sh -- -purge.
```

Distributed Database Deployment

- > Distributed Database Deployment
- > Sizing Requirements for Distributed Database Deployment
- > Install Cortex XSOAR for a Distributed Database Deployment
- > Install a Distributed Database Node
- > Configure Live Backup for a Distributed Database Deployment
- > Change the Node Admin Password
- > Delete a User from a Node
- > Convert a Single Server Deployment to a Distributed Database Deployment
- > Upgrade the Cortex XSOAR Server for a Distributed Database

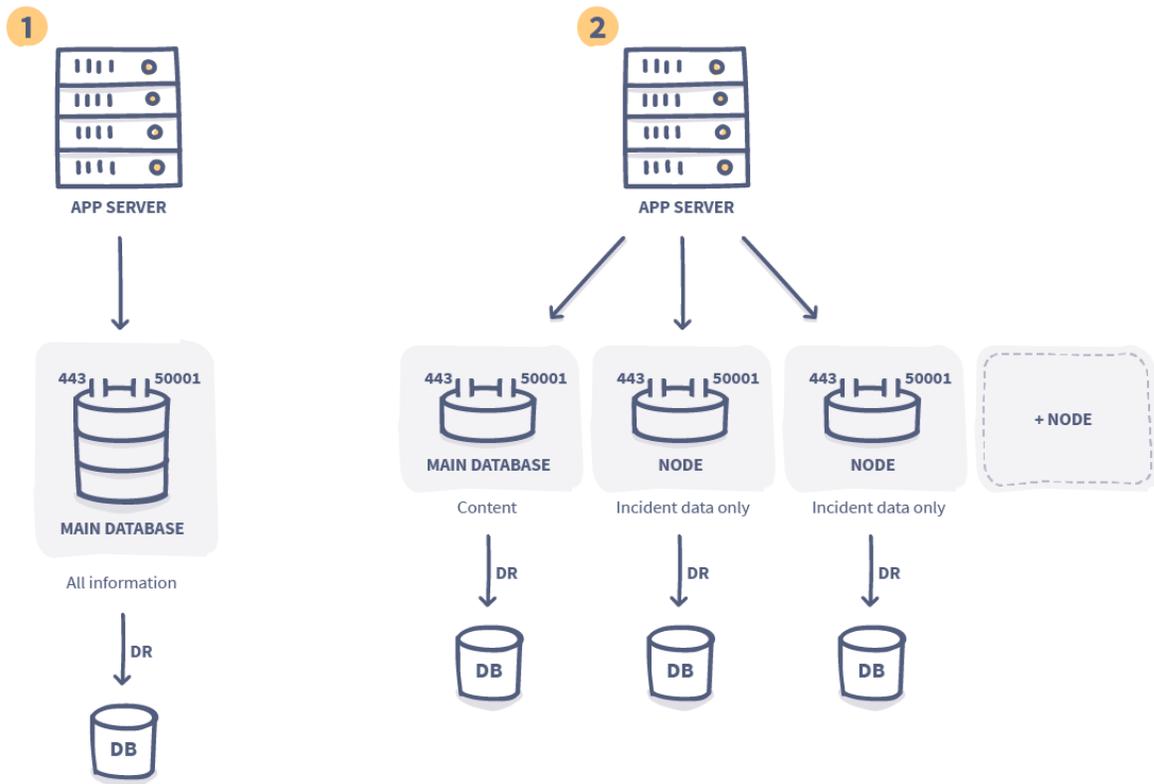
Distributed Database Deployment

This multi-tier configuration enables you to scale your environment and manage load resources. Cortex XSOAR supports two types of multi-tier configurations. In both multi-tier configuration types, there is a single app server.

Each database server, main and nodes, must have its own disaster recovery server configured.

 Although a distributed database deployment might enhance performance, there are various factors that must be considered. This might not be the preferred deployment method for you. Contact your Cortex XSOAR Customer Success manager before you implement a distributed database deployment.

Distributed Database



Single database server

When you deploy Cortex XSOAR with a single app server and a single database server, the database server is considered the main database, on which all content is stored.

Multiple database servers

When you deploy Cortex XSOAR with a single app server and multiple database servers, the first database you install is considered the main database and all additional database servers you install are considered secondary nodes.

The main database server maintains all content that is not an incident or related to an incident, such as playbooks, automations, integrations, and so on.

The nodes maintain all incidents and their related data, for example, the playbook used when processing the specific incident or indicators extracted from the specific incident. Incidents are distributed between the different secondary nodes using a round-robin system.



- *You must ensure that ports 443 and 50001 are open from the app server to the database servers. In addition, port 443 needs to be open while you are initially registering a database node.*
- *Each database server, main and nodes, must have its own disaster recovery configured.*

Sizing Requirements for Distributed Database Deployment

App server

Component	Dev Environment Minimum	Production Minimum
CPU	8 CPU cores	16 CPU cores
Memory	16 GB RAM	32 GB RAM
Storage	500 GB SSD	500 GB SSD

Main database

Component	Dev Environment Minimum	Production Minimum
CPU	8 CPU cores	16 CPU cores
Memory	16 GB RAM	32 GB RAM
Storage	500 GB SSD	500 GB SSD with minimum 3k dedicated IOPS

Database Node

Component	Dev Environment Minimum	Production Minimum
CPU	8 CPU cores	16 CPU cores
Memory	16 GB RAM	32 GB RAM
Storage	500 GB SSD	1 TBSSD with minimum 3k dedicated IOPS

Install Cortex XSOAR for a Distributed Database Deployment

Make sure you meet all sizing requirements for a distributed database deployment.

This document provides information for installing a new Cortex XSOAR environment in a distributed database configuration where the database and app server are installed on different machines.

If you already have a standard Cortex XSOAR deployment, [you can convert it to a distributed database.](#)

STEP 1 | To install the database server, run the `sudo ./demistoserver-X.sh -- -db-only -db-secret=<your_db_secret> -y` command.

Parameter	Description
demistoserver-X	The name of the Cortex XSOAR installer, where X is the version and build number.
db-only	The flag indicating that only the database server is installed.
db-secret	The 10-character string that you defined when you installed the database server.
y	The flag that completes the installation silently by answering yes to the remaining installation questions. Default settings are applied where applicable.

```
sudo ./demistoserver--5.5-45985.sh -- -db-only -db-secret=9876543210 -y
```

STEP 2 | To install the app server, run the `sudo ./demistoserver-X.sh -- -server-only -db-secret=<your_db_secret> -db-address=<IP or hostname> -external-address=<IP or hostname> -y` command.

Parameter	Description
demistoserver-X	The name of the Cortex XSOAR installer, where X is the version and build number.
server-only	The flag indicating that only the application server is installed.
db-secret	The 10-character string that you defined when you installed the database server.
db-address	The database server's public IP address or hostname. Do not include the http or https prefix.
external-address	The app server's public IP address or hostname. Do not include the http or https prefix.
y	The flag that completes the installation silently by answering yes to the remaining installation questions. Default settings are applied where applicable.

```
sudo ./demistoserver--5.5-45985.sh -- -server-only -db-secret=9876543210 -db-address=2.2.2.2 -external-address=3.3.3.3 -y
```

STEP 3 | (Optional) Install additional database servers (nodes).

Install a Distributed Database Node

When working in a distributed database configuration, you can have multiple databases working with the Cortex XSOAR app server, known as database nodes. The database nodes connect to the Cortex XSOAR application server and provide additional database bandwidth to the main database.

Prerequisite

Go to **Settings > About > Troubleshooting** and verify that the **External Host Name** is populated.

STEP 1 | Go to **Settings > Advanced > Remote Databases**.

STEP 2 | Click **Create Node**.

The installer begins to download.

STEP 3 | Copy the `demistonode-5.0-x.sh` file to the machine on which you want to install the node.

STEP 4 | Run the node installation using the following command: `sudo ./demistonode-X.sh -- -external-address=<IP or hostname> -y`.

Parameter	Description
demistonode-X	The name of the Cortex XSOAR database node installer, where X is the version and build number.
external-address	The node's public IP address or hostname. This is the address that the node uses to register with to the cluster and that the App Server uses to communicate with the node. Do not include the http or https prefix.
y	Answers all remaining installation questions with a yes (using default settings) and enables you to continue the installation silently.

For example: `sudo ./demistonode--5.0-45985.sh -- -external-address=5.5.5.5 -y`

STEP 5 | In Cortex XSOAR, verify that the new database node appears in the list of remote databases.

Configure a Live Backup for a Distributed Database Overview

Live Backup for distributed database environments enables you to mirror your active database servers to passive servers, and, in a disaster recovery scenario, easily convert your passive servers to the active database servers.



Live Backup only backs up the database servers. As no information is stored on the application servers, there is no need to back those up. In the event that an application server fails, you can [install another application server](#).

Server actions are mirrored in real-time. There may be pending actions due to high server load, connectivity issues, and so on. Note the following:

- Live Backup uses a single active server and a single standby server.
- Active / Active configuration is not currently supported.
- Each host retains its own distinct IP address and hostname.
- Neither host has any awareness of which node is truly active. Therefore, failover is not dynamic, meaning that making a node active must be done manually, by an administrator.

In the event of a server failover, engines dynamically reconnect to the active host.



As the process of making a Cortex XSOAR server active is a manual process, it is conceivable that two servers could be active simultaneously. You must avoid this scenario because both hosts collect and work on potentially the same security incidents, which could possibly lead to the following:

- *Incident duplication.*
- *A higher load on your integration endpoints.*
- *Possible significant database inconsistencies due to duplication of internal identifiers being shared between nodes and causing existing incidents to be overwritten.*



If there is ever uncertainty about whether a host that is presently down or stopped was in an active state before it went offline, it is recommended that you put the presently active node into a standby state before starting the Cortex XSOAR service on the other host. You can then make it active again after you have confirmed whether the host you are starting is already in active mode.

Configure the Live Backup Environment for a Distributed Database

STEP 1 | Navigate to **Settings** > **Advanced** > **Remote Databases**.

STEP 2 | Select the database node for which you want to configure Live Backup and click **Edit Live Backup Configuration**.

STEP 3 | Toggle the Live Backup field to **On**.

STEP 4 | Configure the backup server properties.

1. Property	Value
Hostname	Backup server IP address or hostname (without the https:// prefix).
Port	443 (by default).
Trust server certificate	On: certificates are not checked. Off: certificates are checked.
Use proxy	Select whether to use a proxy.

STEP 5 | Click **Save Live Backup Configuration**.

STEP 6 | Copy the installation package to the machines on which you want to install the live backup servers. Ensure that the backup server has a different IP address from the active server.

STEP 7 | Install Cortex XSOAR using the following command: `./demistoserver-xxxx.sh -- -dr -do-not-start-server -db-only -external-address=<IP or hostname> -db-secret=<your_db_secret> -y`

2. Parameter	Description
demistoserver-X	The name of the Cortex XSOAR installer, where X is the version and build number.
dr	The flag indicating that the installation is for disaster recovery.
do not start server	The flag indicating that the server should not be started.
db-only	The flag indicating that only the database server is installed.
db-secret	The 10-character string that you defined when you installed the database server.
external-address	The app server's public IP address or hostname.  <i>Do not include the http or https prefix.</i>
y	Answers all remaining installation questions with a yes (using default settings) and enables you to continue the installation silently.  <i>This automatically enables telemetry. For more information, see Cortex XSOAR Telemetry.</i>

For example: `sudo ./cortexxsoarserver--5.5-70066.sh -- -dr -do-not-start-server -db-only -db-secret=9876543210 -external-address=2.2.2.2 -y`

STEP 8 | Verify that the passive server is accessible from the active server through port 443 (or any other port configured as a listening port). Make sure that there are no firewalls that might drop communication.

STEP 9 | Stop the Cortex XSOAR server.

STEP 10 | Create a tarball file of the following necessary files and folders on the active server to be copied to the passive server.

- /var/lib/demisto/data
- /var/lib/demisto/artifacts
- /var/lib/demisto/attachments
- /var/lib/demisto/systemTools
- /var/lib/demisto/d2_server.key
- /usr/local/demisto/cert*
- /usr/local/demisto/demisto.lic

To create the file, use the following command, which preserves demisto:demisto ownership and file permissions. `tar --ignore-failed-read -pczf demistoBackup.tgz /var/lib/demisto/data /var/lib/demisto/artifacts /var/lib/demisto/attachments /var/lib/demisto/systemTools /var/lib/demisto/d2_server.key /usr/local/demisto/cert* /usr/local/demisto/demisto.lic`

STEP 11 | Copy the created tarball file (demistoBackup.tgz) to the passive server using either `scp` or a tool that you prefer. For example, `scp demistoBackup.tgz root@<yourBackupServerIPorHostname>:/root`

STEP 12 | On the passive server, extract the backup tarball file with the following command (original file permissions and ownership will be preserved): `tar -C / -xzipvf demistoBackup.tgz`

STEP 13 | Start the passive server.

STEP 14 | Start the active server.

If the procedure was successful, you will see the following information populated in the table in **Settings > Advanced > Remote Databases**.

Property	Value
DR Link	The hostname or IP address of the backup server.
DR Status	The connection status to the backup server. Possible values are: <ul style="list-style-type: none">• Ok• Pending• Error

Transition a Standby Server to Active Mode

The following instructions are for replacing the main database in the event of a real disaster scenario. This means that the main database is down and cannot be brought back online. Since the user interface is not available, the procedure involves manual steps.

STEP 1 | Manually verify that the main database really is down (as opposed to a connectivity issue for a particular app server). Make sure it will not come back online later.

1. Determine if the main database's host machine is down and cannot be restarted.
2. If the main database's host machine is up, log into it and check if the demisto service is down and cannot be restarted.
3. If the demisto service is up, rule out a network or firewall issue by checking if the main database's machine is accessible from the app server's machine.

STEP 2 | For each app server:

1. Stop the service.
 - CentOS: `sudo systemctl stop demisto`
 - Ubuntu: `sudo service demisto stop`
2. Manually change the active database address in the server's local configuration file. The default location of the configuration file is `/etc/demisto.conf`

For example:

```
{
  "Server": {
    "ExternalHostName": "app.server.com",
    "HttpsPort": "443",
    "appID": "f4ea5263-906a-4ab8-83d4-f9f64563926c",
    "inClusterHostName": "app.server.internal",
    "remote": {
      "db": {
        "config": {
          "anycertificate": false,
          "grpcport": "50001",
          "hostname": "main.db.server.com",
          "httpsport": "443",
          "secret": "{TOMMED}pRX3nITZNGGFbwZBScs31BK1Bo4iWh83JS7mRCh",
          "useproxy": false
        }
      },
      "type": "server"
    }
  },
  "db": {
    "index": {
      "entry": {
        "disable": true
      }
    }
  }
}
```

3. Start the service.
 - CentOS: `sudo systemctl start demisto`
 - Ubuntu: `sudo service demisto start`

STEP 3 | Navigate to the URL of the passive server and log in as an administrator. You are presented with the following page:



STEP 4 | Click the **Make this the production server** button.

STEP 5 | In the Switch Hosts dialog window, type `switch hosts` in the text box.

The database monitoring dashboard should appear and the database server is now active.

When the system is back online, you will probably want to set up a new standby server for your main database. See [Configure a Live Backup for a Distributed Database Overview](#) for details.

Transition an Active Server to Standby Mode for a Distributed Database

This procedure assumes that your main database is online. If you are dealing with a failure of the main database, follow the instructions in [Transition a Standby Server to Active Mode](#).

STEP 1 | On the application server, navigate to **Settings > Advanced > Remote Databases**.

STEP 2 | Select the database server you want to transition, and click **Switch Host**.

STEP 3 | In the Switch Hosts dialog window, type `switch hosts` in the text box.

STEP 4 | Click the **Yes, switch hosts** button to commit the change.



It may take several minutes for the switch to occur and be reflected in the page.

If you used this procedure to recover from a database failure, you will probably want to set up a new standby server for that database. See [Configure a Live Backup for a Distributed Database Overview](#) for details.

Change the Node Admin Password

By default, after installing a node, the password for the database server is the same as the admin user and password on the App server. For security reasons, you might want to change this password.

STEP 1 | Open a command line window.

STEP 2 | Make the following API call.

```
curl --request POST \ --url https://<SERVER>:<PORT>/users/setpw \ --header 'Remote-Database-Authorization: <Base64-DB-SECRET>' \ --header 'content-type: application/json' \ --data '{ "id": "<userName>", "password": "<newPassword>" }'
```

Parameter	Description
Server	The hostname or IP address of the database server node.
Port	The port through which you are connecting. Do not include the http or https prefix.
Remote-Database-Authorization	The header you need to provide to circumvent user authentication.
DB-SECRET	The 10-character string that you defined when you installed the main database server.  <i>When sending the API call, make sure that the string you insert is encoded as base-64.</i>
id	The name of the user whose password you want to change.
password	The new password for the admin user.

Delete a User from a Node

Follow these steps to delete a user from a node.

STEP 1 | Open a command line window.

STEP 2 | Make the following API call.

```
curl --request POST \ --url https://<SERVER>:<PORT>/users/delete \ --header 'Remote-Database-Authorization: <base64-DB-SECRET>' \ --header 'content-type: application/json' \ --data '{ "ids": ["userID"]}'
```

Parameter	Description
Server	The hostname or IP address of the database server node.
Port	The port through which you are connecting. Do not include the http or https prefix.
Remote-Database-Authorization	The header you need to provide to circumvent user authentication.
DB-SECRET	The 10-character string that you defined when you installed the main database server.  <i>When sending the API call, make sure that the string you insert is encoded as base-64.</i>
id	The name of the user to delete from the node. To get a list of use IDs, make a users/delete call and leave the ids field empty.

Convert a Single Server Deployment to a Distributed Database Deployment

This information is only intended for existing, single-server deployments that you want to convert to a distributed database deployment.

To ensure that you do not lose your existing data, the machine on which Cortex XSOAR is currently installed must be converted into the database server.



Your Cortex XSOAR Engines are currently configured to connect to the previous configuration of the Cortex XSOAR platform, which you have now converted to the database server. You need to change the engine configuration so that it connects to the newly installed Cortex XSOAR application server.

STEP 1 | Convert the existing deployment.

1. Stop the server using the `sudo service demisto stop` command.
2. **Optional** If you have docker images that you want to copy, run the following command:

```
docker save -o <name>.tar <image>
```

Where:

- name is the name of the output file to which you want to save the docker image
- image is the current docker image that you want to save

Copy the tar file you created to the machine on which you will install the application server.

3. Run the Cortex XSOAR installation package using the `sudo ./demistosever-X.sh -- -db-only -db-secret=<your_db_secret> -y` command.

Parameter	Description
demistosever-X	The name of the Cortex XSOAR installer, where X is the version and build number.
db-only	The flag indicating that only the database server is installed.
db-secret	A 10-character string that you will need to provide for the app server.
y	Answers all remaining installation questions with a yes (using default settings) and enables you to continue the installation silently.

For example: `sudo ./demistosever-5.0-45985.sh -- -db-only -db-secret=9876543210 -external-address=2.2.2.2 -y`

- ## STEP 2 | Install the Cortex XSOAR app server using the `sudo ./demistosever-X.sh -- -server-only -db-secret=<your_db_secret> -db-address=<IP or hostname> -external-address=<IP or hostname> -y` command.

Parameter	Description
demistosever-X	The name of the Cortex XSOAR installer, where X is the version and build number.
server-only	The flag indicating that only the appserver is installed.
db-secret	A 10-character string that you will need to provide for the app server.
db-address	The database server's public IP address or hostname. Do not include the http or https prefix.
external-address	The app server's public IP address or hostname. Do not include the http or https prefix.
y	Answers all remaining installation questions with a yes (using default settings) and enables you to continue the installation silently.

For example: `sudo ./demistosever--5.0-45985.sh-- -server-only -db-secret=9876543210 -db-address=2.2.2.2 -external-address=3.3.3.3 -y`

1. **Optional** If you copied docker images from the database server to the application server, run the following command:

```
docker load -i <name>.tar
```

Where name is the name of the file to which you saved the docker image(s).

2. Verify that your docker images were properly loaded using the `docker images` command.

STEP 3 | Reconnect all Cortex XSOAR engines.

1. Connect to the machines on which the engine is installed.
2. Stop the engine service using the `sudo service d1 stop` command.
3. Navigate to `/usr/local/demisto/` and open `d1.conf` in a text editor.
4. Under **EngineURLs** and **AgentURLs**, replace the current values with the app server's external IP address or hostname.
5. Under **ServerPublic**, replace the current value with an updated value.
 - In Cortex XSOAR, go to **Settings > Integrations > Engines**.
 - Click **Create New Engine**.
 - Under **Installer Type**, select **Zip**.
 - Download the newly created engine package.
 - Open the `d1.conf` file located in the engine package in a text editor.
 - Copy the value of the **ServerPublic** field and past it in the `d1.conf` file located on your engine machine.
6. Restart the engine using the `sudo serviced1 start` command.

STEP 4 | Check the status of the engine using the `sudo systemctl status d1` command.

The engine logs are available at `/var/log/demisto/d1.log`

Reindex Databases in a Distributed Database Deployment

The following are the steps required to reindex databases in a distributed database deployment. The amount of time needed to reindex the databases depends on the amount of data in the databases.

STEP 1 | Log in to the app server and databases (main, node1, node2...) via SSH.

STEP 2 | Stop all Cortex XSOAR services in the following order: app server, node databases, main database.

```
sudo service demisto stop
```

STEP 3 | Back up the index directory in each database.

```
cp -r /var/lib/demisto/data/demistoidx /tmp/demistoidx
```

STEP 4 | Remove the index directory by running:

```
rm -rf /var/lib/demisto/data/demistoidx
```

STEP 5 | Start the main database.

```
sudo service demisto start
```

STEP 6 | When the main database is up, start the node databases.

```
sudo service demisto start
```

STEP 7 | Start the app server.

```
sudo service demisto start
```

STEP 8 | Verify that all of the remote databases are enabled.

```
<cortexXSOAR_server>/#/settings/remoteDatabaseMonitoring
```

To enable remote database, go to **Settings > Remote Database** and select the checkbox for the remote database and click **Enable Node**.

Restore Databases in a Distributed Database Deployment

Follow these steps to restore the databases in a distributed database deployment.

STEP 1 | Log in to the app server and databases (main, node1, node2...) via SSH.

STEP 2 | Stop all Cortex XSOAR services in the following order: app server, node databases, main database.

```
sudo service demisto stop
```

STEP 3 | Restore the index directory in each database.

```
sudo cp -r /tmp/demistoidx /var/lib/demisto/data/demistoidx
```

STEP 4 | Change the owner of the databases.

```
sudo chown demisto:demisto /var/lib/demisto/data/demistoidx
```

STEP 5 | Start the main database.

```
sudo service demisto start
```

STEP 6 | When the main database is up, start the node databases.

```
sudo service demisto start
```

STEP 7 | Start the app server.

```
sudo service demisto start
```

STEP 8 | Verify that all of the remote databases are enabled.

```
<yourdemisto.com>/#/settings/remoteDatabaseMonitoring
```

To enable remote database, go to **Settings > Remote Database** and select the checkbox for the remote database and click **Enable Node**.

Upgrade the Cortex XSOAR Server for a Distributed Database

The installer automatically detects the existing configurations and applies them to the upgraded server

STEP 1 | Copy the installer to each of the machines in the distributed database environment.

STEP 2 | Stop the app server using the command `sudo service demisto stop`.

STEP 3 | Stop the database servers using the command `sudo service demisto stop` on the main database and all secondary databases.

STEP 4 | To upgrade the database servers, run the `sudo ./demistoserver-X.sh-- -y` command on the main database and all secondary databases.

STEP 5 | To upgrade the app server, run the `sudo ./demistoserver-X.sh-- -y` on the app server.

STEP 6 | Start the database servers using the command `sudo service demisto start` on the main database and all secondary databases..

STEP 7 | Start the app server using the command `sudo service demisto start`.

Proxy

- > Configure Proxy Settings
- > Use NGINX as a Reverse Proxy to the Cortex XSOAR Server

Configure Proxy Settings

Proxy settings can be configured globally in Cortex XSOAR by adding them as a server configuration.



Generally, when you need a proxy for Cortex XSOAR, you also need a proxy for Docker. For information about how to configure Docker to use a proxy, see the [Docker documentation](#). When using a BlueCoat proxy, ensure you encode the values correctly.

STEP 1 | Select **Settings > About > Troubleshooting > Add Server Configuration**.

STEP 2 | Add one of the following keys and values.

Configuration	Key	Value
http proxy	<code>http_proxy</code>	<code>http://user:password@prox-server:port#</code> For example, <code>http://user:password@prox-server:3128</code>
https proxy	<code>https_proxy</code>	<code>https://user:password@prox-server:port#</code> For example, <code>https://user:password@prox-server:3128</code>

STEP 3 | Click **Save**.

Use NGINX as a Reverse Proxy to the Cortex XSOAR Server

NGINX can act as a reverse proxy that sits between internal applications and external clients, forwarding client requests to the appropriate application. Using NGINX as a reverse proxy in front of the Cortex XSOAR server enables you to provide network segmentation where the proxy can be put on a public subnet (DMZ) while the Cortex XSOAR server can be on a private subnet, only accepting traffic from the proxy. Additionally, NGINX provides a number of advanced load balancing and acceleration features that you can utilize.

The following topics describe how to install, use a Self-Signed Certificate for non-production environments, and how to configure NGINX.

- [Install NGINX on Cortex XSOAR](#)
- [Generate a Certificate for NGINX](#)
- [Configure NGINX](#)

Use Engines Through the NGINX Reverse Proxy

If you want to use a Cortex XSOAR Engine (d1) through the reverse proxy, you need to modify the following entries in the `d1.conf` file to point to the host and port the NGINX server is listening on:

- **EngineURLs**
- **AgentURLs**

Install NGINX on Cortex XSOAR

You can install NGINX on the Red Hat/Amazon (yum) and Ubuntu Linux distributions. For full instructions and available distributions, see [NGINX documentation](#).

STEP 1 | Run one of the following commands according to your Linux system:

- **RedHat/Amazon:** `sudo yum install nginx`
- **Ubuntu:** `sudo apt-get install nginx`

STEP 2 | (Optional) Verify the NGINX installation by running the following command:

```
sudo nginx -v
```

Generate a Certificate for NGINX

You should not use self-signed certificates for production systems. It is recommended to use a properly signed certificate for production systems. These instructions are intended only for non-production setups

STEP 1 | To use OpenSSL to generate a self-signed certificate run the following command:

```
sudo openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout /etc/nginx/cert.key -out /etc/nginx/cert.crt
```

STEP 2 | When prompted, complete the on-screen instructions to complete the required fields.

Configure NGINX

Follow these instructions to configure NGINX on Cortex XSOAR.

STEP 1 | Open the following NGINX configuration file with your preferred editor:

```
/etc/nginx/conf.d/demisto.conf
```

STEP 2 | Use the following configuration template:

Replace **DEMISTO_SERVER** with the appropriate hostname.

```
# Replace DEMISTO_SERVER with the appropriate hostname. If needed, change
port 443 to the port on which the Demisto server is listening.

upstream demisto {
server DEMISTO_SERVER:443;
}

# Uncomment to redirect http to https (optional)
# server {
#     listen 80;
#     return 301 https://$host$request_uri;
# }

server {
# Change the port if you want NGINX to listen on a different port
listen 443;

ssl_certificate             /etc/nginx/cert.crt;
ssl_certificate_key        /etc/nginx/cert.key;

ssl on;
ssl_session_cache builtin:1000 shared:SSL:10m;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;
ssl_prefer_server_ciphers on;

access_log                 /var/log/nginx/demisto.access.log;

location / {

proxy_set_header           Host $host;
proxy_set_header           X-Real-IP $remote_addr;
proxy_set_header           X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header           X-Forwarded-Proto $scheme;

proxy_pass                 https://demisto;
proxy_read_timeout        90;
}

# Note the regex prefix '^(/acc_\w+)?' is needed for multi-tenant
environments
# In non MT envs you can use the following: location ~ /(websocket|
dlws|d2ws)
location ~ ^(/acc_\w+)?(/websocket|dlws|d2ws) {
proxy_pass https://demisto;
proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "upgrade";
proxy_set_header Host $host;
```

```
proxy_set_header Origin "";
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
}
```

STEP 3 | Restart the NGINX server, by typing the following command:

```
sudo service nginx restart
```

STEP 4 | Verify you can access Cortex XSOAR by browsing to the NGINX server host.

Manage Data

- > Reindex the Entire Database
- > Reindex a Specific Index Database
- > Reindex the Entire Database for a Distributed Database
- > Reindex a Specific Index for a Distributed Database
- > Free up Disk Space with Data Archiving
- > Migrate Data to Another Server
- > Move Data Folders to Another Location on the Server
- > Restore an Archived Folder

Reindex the Entire Database

Follow these steps to reindex the entire database.

STEP 1 | Stop the Cortex XSOAR service using the appropriate command for your OS.

- `systemctl stop demisto`
- `sudo service demisto stop`

STEP 2 | Backup the index directory.

```
/var/lib/demisto/data/demistoidx
```

STEP 3 | Delete the index folder using the following command.

```
rm -rf /var/lib/demisto/data/demistoidx
```

STEP 4 | Set permissions using the following command.

```
sudo chown -R demisto:demisto /var/lib/demisto/data
```

STEP 5 | Start the Cortex XSOAR service using the appropriate command for your OS.

- `systemctl start demisto`
- `sudo service demisto start`

STEP 6 | Log in to your Cortex XSOAR instance and verify that the reindex process was successful.

All should appear, for example, incidents, playbooks, automations, and so on.

Reindex a Specific Index Database

Follow these steps to reindex a specific index database. You can also [reindex the entire database](#). Depending on the volume of the data in the system, it may take some time for the indexing to complete. You can reindex more than one database at a time by listing the indexes as a comma separated list

When you are working in a [Distributed Database Deployment](#) distributed database configuration, nodes must be enabled. To enable nodes, go to **Settings > Remote Database** and select the checkbox for the node and click **Enable Node**.

STEP 1 | Stop the Cortex XSOAR service using the appropriate command for your OS.

- `systemctl stop demisto`
- `sudo service demisto stop`

STEP 2 | Backup the index directory using the following command.

```
tar -czvf filename.tar.gz /var/lib/demisto/data/demistoidx
```

The backup of the index directory should be not stored under `/var/lib/demisto`.

STEP 3 | Run the server and be sure to specify the index as an argument.

```
sudo /usr/local/demisto/server -restore-index-name=indexName -public /usr/local/demisto/dist -stdout -conf /etc/demisto.conf
```

STEP 4 | Log in to your Cortex XSOAR instance and verify that the reindex process was successful.

STEP 5 | When a message appears stating `server up and running, good luck to us all` or the Cortex XSOAR UI displays, verify that all the data is present from the reindexing, and stop the process by pressing `Ctrl+C` or `Cmd+C`.

STEP 6 | Set permissions by running the following command.

```
sudo chown -R demisto:demisto /var/lib/demisto/data
```

STEP 7 | Start the Cortex XSOAR service using the appropriate command for your OS.

- `systemctl start demisto`
- `sudo service demisto start`

Reindex the Entire Database for a Distributed Database

Follow these steps to reindex the entire database in a distributed database environment. Depending on the volume of the data in the system, it may take some time for the indexing to complete.

STEP 1 | Stop the Cortex XSOAR service.

```
sudo systemctl stop demisto
```

STEP 2 | Stop the demisto instance on the main database and all secondary databases.

```
sudo systemctl stop demisto
```

STEP 3 | Delete the index folder on all databases using the following command.

```
sudo rm -rf /var/lib/demisto/data/demistoidx
```

STEP 4 | Set permissions on all databases using the following command.

```
sudo chown -R demisto:demisto /var/lib/demisto/data
```

STEP 5 | Start the main database and all secondary databases.

```
sudo systemctl start demisto
```

STEP 6 | Start the Cortex XSOAR service.

```
sudo systemctl start demisto
```

STEP 7 | Log in to your Cortex XSOAR instance and verify that the reindex process was successful.

All of your data should appear, for example, incidents, playbooks, automations, and so on. If there is a problem, contact the Cortex XSOAR support team.

Reindex a Specific Index for a Distributed Database

Follow these steps to reindex a specific index in a distributed database environment. You need to log in to each database machine and run the indexing command with the name of the specific index on each database machine. Depending on the volume of the data in the system, it may take some time for the indexing to complete.

STEP 1 | Stop the Cortex XSOAR service.

```
sudo systemctl stop demisto
```

STEP 2 | Log in to the main database machine and all secondary database machines.

STEP 3 | Stop the demisto instance on the main database and all secondary databases.

```
sudo systemctl stop demisto
```

STEP 4 | On each database machine, run the server and be sure to specify the index as an argument.

```
sudo /usr/local/demisto/server -restore-index-name=indexName -public /usr/local/demisto/dist -stdout -conf /etc/demisto.conf
```

STEP 5 | Log in to your Cortex XSOAR instance and verify that the reindex process was successful.

STEP 6 | When a message appears stating **server up and running, good luck to us all** or the Cortex XSOAR UI displays, verify that all the data is present from the reindexing, and stop the process by pressing **Ctrl+C** or **Cmd+C**.

STEP 7 | Set permissions on all databases using the following command.

```
sudo chown -R demisto:demisto /var/lib/demisto/data
```

STEP 8 | Start the main database machine and all secondary database machines.

```
sudo systemctl start demisto
```

STEP 9 | Start the Cortex XSOAR service.

```
sudo service demisto start
```

Free up Disk Space with Data Archiving

Cortex XSOAR supports full archiving of incidents, entries and indicators by month. Data from incidents, insights (indicators), and entries are stored in folders on a monthly basis.

If disk space needs to be freed up, you can archive folders to condense unused data within them. It is recommended to archive folders and not delete them permanently.

Locate the folders that reside in the following location (where Cortex XSOAR is installed), `/var/lib/demisto/data/`.



*Although the folders reside in `/var/lib/demisto/data/`, **Do Not** save the backup folders under `/var/lib/demisto/`.*

The following data folder and files can be found in this folder.

- `demisto.db` - db for all playbooks and automation - all things not having to do with incidents and insights
- `demistoidx` - indexing of the system
- `partitionsData` - Data of incidents, insights, entries split up by month resolution

The following is an example of how the folders and filenames will appear in your system.

```
$ tree /var/lib/demisto/data
### demisto.db
### demistoidx
#   ### accounts
#   #   ### index_meta.json
#   #   ### store
...
#   ### entries_082017
#   #   ### index_meta.json
#   #   ### store
#   ### entries_092017
#   #   ### index_meta.json
#   #   ### store
#   ### entries_102017
#   #   ### index_meta.json
#   #   ### store
#   ### evidences
#   #   ### index_meta.json
#   #   ### store
#   ### incidents_082017
#   #   ### index_meta.json
#   #   ### store
#   ### incidents_092017
#   #   ### index_meta.json
#   #   ### store
#   ### incidents_102017
#   #   ### index_meta.json
#   #   ### store
#   ### investigations_082017
#   #   ### index_meta.json
#   #   ### store
#   ### investigations_092017
#   #   ### index_meta.json
```

```
# # ### store
# ### investigations_102017
# # ### index_meta.json
# # ### store
...
# ### newInsights_082017
# # ### index_meta.json
# # ### store
# ### newInsights_092017
# # ### index_meta.json
# # ### store
# ### newInsights_102017
# # ### index_meta.json
# # ### store
# ### playbooks
# # ### index_meta.json
# # ### store
...
### partitionsData
### demisto_082017.db
### demisto_092017.db
### demisto_102017.db
```

Follow these steps to restore a folder that was previously archived.

STEP 1 | Stop the Cortex XSOAR service using the following command.

```
$ sudo service demisto stop
```

STEP 2 | Navigate to the `/var/lib/demisto` filepath using the following command.

```
cd /var/lib/demisto
```

STEP 3 | Create a directory named `archive` using the following command.

```
mkdir archive
```

STEP 4 | Move the data you want to archive to the archive directory using the following command.

```
mv data/**/*_<file_name>* archive
```

STEP 5 | Create the compressed archive of your selected files and folders using the following tarball command.

```
$ tar -cvzf demisto-<file_name>-archive.tar.gz archivekdir a
```

STEP 6 | Start the Cortex XSOAR service using the following command.

```
sudo service demisto start
```

Migrate Data to Another Server

In certain cases you might need to move data between servers, for example, when a stronger production server is required.

STEP 1 | Install Cortex XSOAR on the new server.

Do not start the server when the installation finishes.

STEP 2 | Change the `/var/lib/demisto` directory to `/var/lib/demisto.old`.

STEP 3 | Copy the following files and directories from the old server to the new server.

- `/var/lib/demisto`
- `cert.key` and `cert.pem` under `/usr/local/demisto`

Make sure that ownership of the directories and file are set to **demisto:demisto**

STEP 4 | Start the new server and wait for the server to complete the indexing process.

Move Data Folders to Another Location on the Server

Follow these steps to move the directories installed on the Cortex XSOAR server to another location.

Let *<new path>* mark the base path to which to move the Cortex XSOAR server files.

Always use the `cp` command with the `-p` flag to retain the original file permissions.

Use the following commands to only move the data.

STEP 1 | Stop the Cortex XSOAR service.

```
sudo service demisto stop
```

STEP 2 | Move `/var/lib/demisto` to `<new path>/var/lib/demisto` .

```
cp -pfr /var/lib/demisto <new path>/var/lib
```

STEP 3 | Add or modify keys by editing the existing `/etc/demisto.conf` file.

Contents of the `/etc/demisto.conf` file are in JSON format and can hold additional data than what is shown in the example. In most cases, the existing `demisto.conf` file will not contain the necessary keys, such as *folders* and *DB*.

```
{
  "Server": {
    "HttpsPort": "443"
  },
  "folders": {
    "lib": "{new path}/var/lib/demisto"
  },
  "DB": {
    "DatabaseDir": "{new path}/var/lib/demisto/data",
    "IndexDir": "{new path}/var/lib/demisto/data"
  }
}
```

STEP 4 | Start the Cortex XSOAR service using the appropriate command for your OS.

```
sudo service demisto start
```

STEP 5 | (Optional) Verify that the process was performed successfully and remove the original directory `/var/lib/demisto`.

Restore an Archived Folder

Follow these steps to restore a folder that was previously archived.

STEP 1 | Stop the Cortex XSOAR service using the appropriate command for your OS.

- `systemctl stop demisto`
- `sudo service demisto stop`

STEP 2 | Navigate to the `/var/lib/demisto` filepath using the following command.

```
cd /var/lib/demisto
```

STEP 3 | Create a directory named `archive` using the following command.

```
mkdir archive
```

STEP 4 | Restore the folder using the following command, where ***folderName*** is the name of the folder to restore.

```
tar -C archive -xvzf folderName
```

STEP 5 | Move the idx data back to the original `demistoidx` folder using the following command.

```
mv archive/*2017 data/demistoidx
```

STEP 6 | Move the partitions back to the original `partitionsData` folder using the following command.

```
mv archive/*2017.db data/partitionsData
```

STEP 7 | Start the Cortex XSOAR service using the appropriate command for your OS.

- `systemctl start demisto`
- `sudo service demisto start`

Users and Roles

- > Users and Roles Overview
- > Roles in Cortex XSOAR
- > Self-Service Read-Only Users
- > Shift Management
- > User Invitations
- > Integration Permissions
- > Password Policy
- > Change the Administrator Password
- > Authenticate Users with SAML 2.0
- > Configure User Notifications
- > Set the Default Theme for New Users

Users and Roles Overview

Cortex XSOAR uses role-based access control (RBAC) for controlling user access. RBAC helps manage access to Cortex XSOAR components, so that users, based on their roles, are granted minimal access required to accomplish their tasks.

You can manage the following settings/roles in the **USERS AND ROLES** tab:

- View and manage different roles and access permissions in the **Roles** tab. You can add as many roles as required and change their permission levels, as described in [Roles in Cortex XSOAR](#).
- View and manage different users in the **Users** tab. You can view the user's details such as name, email address, last log in, whether they have been locked out, and so on. You can also manage the user's password, unlock their account, disable, enable, and remove their account.
- Invite users and manage invitations, as described in [User Invitations](#). After the user has accepted the invitation you can manage their role in Cortex XSOAR.
- Assign roles to commands at the integration instance level. This means if you have multiple instances of the same integration, you can assign different roles (permission levels) for the same command in each instance. For more information, see [Integration Permissions](#).
- View details of actions taken in Cortex XSOAR in the **Audit trail**.
- Set a password policy, as described in [Password Policy](#).

You can also authenticate users with SAML 2.0, using Okta, Azure, and so on, as described in [Authenticate Users with SAML 2.0](#).

Roles in Cortex XSOAR

A role is a set of permissions that determine which actions and resources users within that role are granted access in Cortex XSOAR. Users are assigned to at least one role, depending on their required level of access.

Roles

Cortex XSOAR has the following assigned roles:

Role	Default Permissions
Administrator	Read/Write permissions for all components and access to all pages.
Analyst	Mix of Read and Read/Write permissions for all components and access to all pages.
Read-Only	Read permissions for all components and access to all pages.

You can add as many roles as you require, by clicking **New**. To create a new role, see [create a new role](#). Follow the same steps when editing a role. When defining a new role, you can add permissions, SAML and AD Roles, defining shift periods and so on.

Permissions

You can view and change the following permission levels as required:

Permission	Description
None	No access to the specified component.
Read	Can view but not edit the specified component.
Read/Write	Can view and edit the specified component.

Shifts

If you want to manage shift periods for users, including who is on call and to whom to assign, you can [define a role](#) for a specific shift period and then assign that shift to a user.

Define a Role

Cortex XSOAR comes with three roles with default permissions. You can add as many new roles and combine them with other roles, such as single sign on.

STEP 1 | In the **Roles** tab, click **New**.

STEP 2 | In the **Role name** field, type the name for the new role.

STEP 3 | Select the category permissions.

Component	Description
Investigations	Sets the permission level generally for investigations or set different permission levels for data and chats. You can also limit the role to exclude executing potential harmful actions when building your own integrations.
Incident table actions	Limits table actions in the Incidents page, such as delete, edit, close and so on.
Jobs	Limits permissions for managing jobs.
Scripts	Limits permissions for managing scripts. If the user has read/write permissions you can enable users to create scripts that run as a Super User. In the Script page, you can define which roles are permitted to run an automation, and according to which role the automation executes.
Playbooks	Limits permissions for creating, editing and deleting Playbooks. You can also add, change, and remove roles from a playbook when clicking Settings in the Playbooks page. There are several notes and limitations you should familiarize yourself with when assigning roles to playbooks.
Settings	You can set the permission level generally for all settings or split them according to the following: Users: includes invitations and editing permissions. Integrations: whether a user can add, edit or delete instances. Credentials: whether a user can add, edit, or delete credentials.
Administration	Limits permissions for server configurations, editing layouts for indicators and incidents, integration permissions, audit trails and the password policy.

STEP 4 | In the **Page Access** section, select the pages you want the user to have access.

STEP 5 | To assign the role to an active directory group, in the **AD Roles Mapping** section, from the drop down list, select the group as required.

STEP 6 | To assign a role to a single sign on group, in the **SAML Roles Mapping** section, from the drop down list, select the group as required.

To associate roles to an AD or SAML group, you need to add a SAML instance and configure your identity provider.

Users can log into Cortex XSOAR with their Active Directory or SAML user name and passwords. Their permission in Cortex XSOAR is set according to the groups and mapping set in Active Directory or SSO. For more information, [Authenticate Users with SAML 2.0](#).

STEP 7 | If you want to associate the role with another role, in the **Nested Roles** section, from the drop down list, select the nested role, as required.

The **Nested Role** overrides any settings you select in the **Roles** tab.

STEP 8 | To add a shift period of work to the role, in the **Shifts** field, click **+ Add Shift** and define the required period.

Weekly shifts start on Sunday and specified in the UTC time zone.

STEP 9 | Click **Save**.

Default Admin

You can configure a user with an administrator role to be a default admin user. Default admin users are not counted as license users, cannot be deleted, and is also a tenant admin.

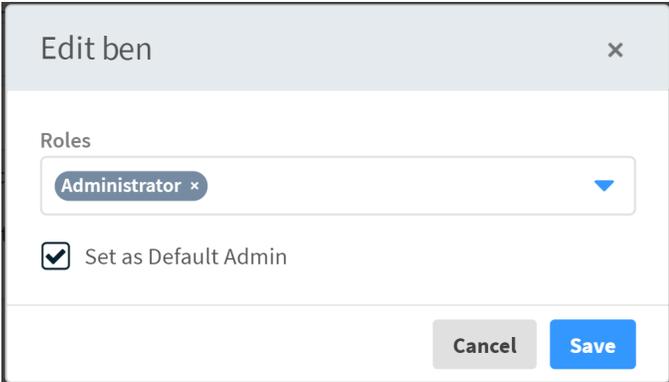
A default admin user has the following privileges:

- Users and Roles
 - Change the role of users
 - View all users' invitations
 - View all users
 - Enable/disable other users
- API Keys
 - Revoke all API keys
 - Create/update API keys for other users
- Incidents
 - Change any incident field
 - Edit and view any incident
 - View audit log of incidents
- Integrations - Trigger integration fetch command
- Dashboards - Unshare/delete dashboards
- Automation - Run any automation
- Playbooks
 - Change playbook task assignees and complete an task
 - View all scripts/playbooks/all data
- Multi-tenant - View all multi-tenant tenants
- File Entries - Delete files entries from the file system.

STEP 1 | Go to **Settings > Users and Roles > Users**.

STEP 2 | Select a user who has an Administrator role and click **Roles**.

STEP 3 | Select **Set as Default Admin** and click **Save**.



The screenshot shows a dialog box titled "Edit ben" with a close button (X) in the top right corner. Below the title bar, there is a section labeled "Roles" containing a dropdown menu with "Administrator" selected and a small "x" icon to its right. Below the dropdown, there is a checkbox labeled "Set as Default Admin" which is checked. At the bottom right of the dialog, there are two buttons: "Cancel" and "Save".

Self-Service Read-Only Users

The self service read-only users feature provides users who do not have an account and at least one role mapped in Cortex XSOAR the ability to access Cortex XSOAR in a very limited capacity.

Self service read-only users can:

- create incidents
- view their own incidents
- add notes and attachments to their incidents
- view the dashboards created for them by the administrator

An example of an incident that a self service read-only user could create is to report that they lost their laptop.

Self-service read-only users can only view their own data. They cannot start an investigation, create dashboard or reports, or change anything in incidents they create.

In order to create notes, the self service read-only user must mark the **Mark as a note** option.

It is recommended, but not required, that read-only users have an existing account in the company's enterprise directory and Cortex XSOAR configured to authenticate and authorize read-only users using the same the same enterprise directory with LDAP, AD, or SAML authentication protocols.

A user is considered as a read-only user if it has no role associated with the Cortex XSOAR users settings.

To enable the self service read-only user feature, Cortex XSOAR administrators need to:

- Set server configuration parameters to:
 - Allow authenticated users without roles to access the home page.
 - Define the list of dashboards such users have access to.
- Create self service read-only incident types. Since self service read-only users cannot initiate an investigation, the playbooks associated with these incident types should run automatically
- Create self service read-only users if no enterprise directory is configured with Cortex XSOAR.
- Create incident layouts for self service read-only users and allow read-only users to access the incidents tabs containing such layouts.
- Create and share dashboards for read-only users.

Configure the Server for Self Service Read-Only Users

To configure the server for self service, set the **create.read.only.users** and the **dashboards.read.only.users** keys.

STEP 1 | Go to **Settings > About > Troubleshooting**

STEP 2 | In the Server Configuration section, click **Add Server Configuration**.

STEP 3 | Enter the key **create.read.only.users** and value **true**.

STEP 4 | Click **Save**.

STEP 5 | In the Server Configuration section, click **Add Server Configuration**.

STEP 6 | Enter the key **dashboards.read.only.users** and type a comma-separated list of the names of the dashboards you created for the self service read-only users.

STEP 7 | Click **Save**.

Create the Self Service Read-Only Users

If you have an enterprise directory and central authentication is configured on Cortex XSOAR, and if the Self Service Read-Only user configuration is set, then you do not need to do anything to create the user. Any user logging into Cortex XSOAR with a valid enterprise username and password, but without any role associated in Cortex XSOAR's user's configuration, will be treated as a Read-Only user and will enter into the Cortex XSOAR Self Service portal.

However, if there is no enterprise directory and central authentication, then you need to create such users. To do this, create a temporary role and assign it to the user. After you create the user, delete the role, so the user will appear in Cortex XSOAR with no role.

In either case, the read-only users do not count as a user in your license pool.

STEP 1 | Go to **Settings > Users and Roles**.

STEP 2 | Click the **Roles** tab.

STEP 3 | Click **New**.

STEP 4 | In the Role Name field, type the name for the new role. you do not need to assign it any permissions.

STEP 5 | Click the **Users** tab.

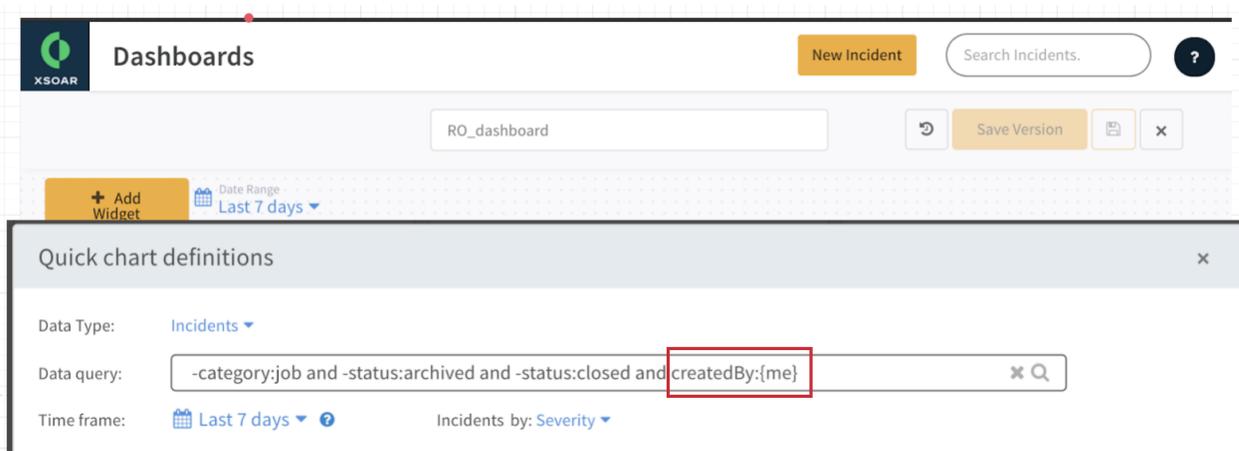
STEP 6 | Click **Invite User**.

STEP 7 | Enter the user's email address and select the role you created and click **Invite**.

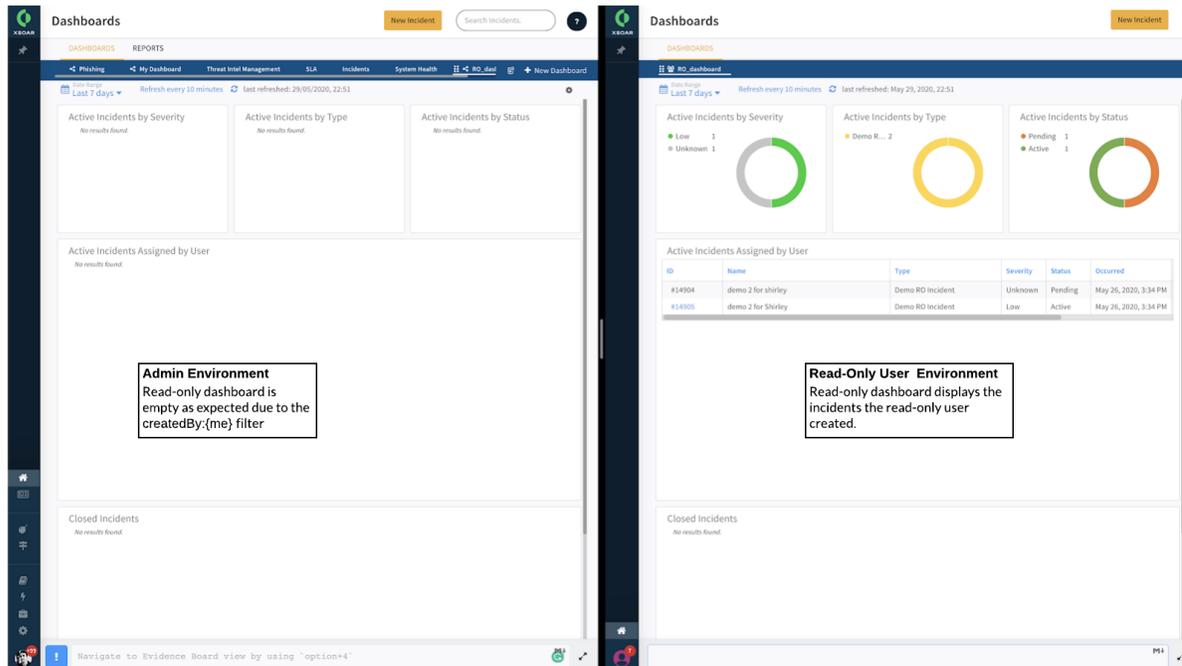
STEP 8 | Click the **Roles** tab and delete the role you created.

Create the Read-Only Dashboard

When creating the dashboard, in the Data Query field of the Quick Chart Definitions window, make sure you add the filter `CreatedBy:me`.



This allows the user to see his own data in the dashboard. The administrator can see the dashboard but cannot see the data.



Be sure to share the dashboard with the read-only user after you create it. For information about creating dashboards, see [Create a New Dashboard](#).

Create the Read-Only Incident Type and Layout

STEP 1 | Go to **Settings > Advanced > Incident Type**.

STEP 2 | Create an Incident Type for the self service read-only user. Be sure to configure the playbook to run automatically.

STEP 3 | Select the Incident Type and click **Edit Layout**.

STEP 4 | Create a new incident tab dedicated to read-only users.

STEP 5 | Hover over the tab you want the self service read-only user to view and click the gear icon.

STEP 6 | Click **Viewing Permissions**.

STEP 7 | In the Viewing Permissions window, select the **Permit users with no roles to view this tab** option.

STEP 8 | Customize this incident tab to only include widgets with data read-only users can access. Such information can be Case Details, Timeline Information, Attachments, and War Room entries.

STEP 9 | Test and validate this incident tab layout as a read-only user to ensure it displays correctly and without an insufficient permission error message.

User Settings and Preferences

You can configure several settings and preferences to customize your work environment.

To view or adjust your user settings and preferences, click your user name on the bottom of the left side toolbar.

Messages

The **Messages** tab is where you view all messages, notifications, incidents, and tags you have been mentioned in. The date is located to the right of each message. At the top of the page is a search bar where you can search for specific messages, notifications, and more.

Details

The **Details** tab is where you update your personal information, including the email address, phone number, and password.

Option Name	Action
Full Name	The display name used in Cortex XSOAR.
Email	The email address to which to receive notifications from Cortex XSOAR.
Phone Number	The phone number where you can be contacted.
Password	Your password for Cortex XSOAR. The password must adhere to these minimum requirements: 8 characters, 1 lowercase letter, 1 uppercase letter, and 1 digit or symbol.
Retype Password	The password you created.
Drop any image here (or click to browse)	The image to use in Cortex XSOAR.

Preferences

The **Preference** tab is where you customize your personal experience with Cortex XSOAR.

Option Name	Action
Default Landing Page	The page to load by default, when logging in to Cortex XSOAR. Can be, Incidents, Dashboard, Playbooks or Automation.
Help Snippet	Whether to have Help Snippets displayed or not.
Enable Shortcuts	Whether to have shortcuts available.
Choose Theme	The preferred theme to be used.

Option Name	Action
Script Editor Style	The style of the editor to use to use in the Automation and BYOI.
Date Format	Your preferred date format and time display format.
Display Timezone	The timezone to associate with your account, in Country/City format.
Highlight Indicators in Playground	Whether to have indicators highlighted in the Playground.
Sign me out of all other sessions (this session remains open)	When clicked, your account will be automatically signed out of all sessions it is signed in to, except, the current session which will remain signed in and open.
Sign me out of all sessions	When clicked, your account will be automatically signed out all active sessions you are logged in to including the current session which will be signed out and will close.

Notifications

The **Notifications** tab enables you to select which notifications to receive and the method for receiving the notifications (email, mobile, or Slack).

Shift Management

Shift management helps you define multiple shifts within Cortex XSOAR. Each shift can be assigned to a user role so you are able to assign one or more analysts across different shifts.

You can do the following:

- Enable incidents to be routed automatically to analysts based on shifts, workloads and machine learning recommendations, ensuring full staff coverage for incoming incidents.
- Define multiple shifts, which can be added to a role, and in turn assigned to a user. To manage shift periods for users, see [Managing Shifts](#).
- Automatically route incidents to analysts based on shifts, load and machine learning recommendations in playbooks and automations

After assigning the role to users, Cortex XSOAR recommends who to assign incidents. When assigning an analyst to an incident, these shifts can be taken into account.



If you want to consider on-call users only run the `getOwnerSuggestions` command.

Managing Shifts

You can define shifts for various roles and then add them to automations, playbooks and incidents.

STEP 1 | Define a Role.

STEP 2 | In the **Shifts** field, click **Add Shift** and add the required period.

Weekly shifts start on Sunday and are specified in the UTC time zone format.

For example, we create a role called First Shift and add a shift starting on Sunday and ending Monday.

Start Day	Start Time	End Day	End Time
Sunday	16:00	Monday	08:00

STEP 3 | Click **Save**.

STEP 4 | (Optional) To add the role to a user, in the **Users** tab, select the user you want to add.

	Roles	Reset P/W	Unlock	Remove	Enable	Disable
<input type="checkbox"/>	Username	Name				Email
<input type="checkbox"/>	John	John				John@demisto.com
<input checked="" type="checkbox"/>	Steven	Steven				Steven@demisto.com
<input type="checkbox"/>	admin	Admin				admin@company.com
<input type="checkbox"/>	richard	richard				richard@demisto.com

STEP 5 | Click **Roles**.

STEP 6 | From the drop down list select the role you created.

Edit Steven ×

Roles

Analyst × FirstShift ×

- Administrator
- Read-Only
- SecondShift

STEP 7 | Click **Save**.

You can also assign the role to a playbook or automation as required.

STEP 8 | (Optional) You can view on-call users details in a dashboard by adding the required widgets.

Dashboards

DASHBOARDS
REPORTS

Incidents
Threat Intel Management
System Health
My Dashboard
SLA
Shift Management

Date Range Last 7 days Refresh every 10 minutes
last refreshed: May 4, 2020, 2:18 PM

Roles Per Shift

Hours / Days	Sunday	Monday	Tuesday	Wednesday	Thursday
0:00 - 1:00		FirstShift	SecondShift		
1:00 - 2:00		FirstShift	SecondShift		
2:00 - 3:00		FirstShift	SecondShift		
3:00 - 4:00		FirstShift	SecondShift		
4:00 - 5:00		FirstShift	SecondShift		

1

Number Of Users On-Call

On-Call Users

Username	Email	Name	Phone	Roles
John	John@demisto.com	John		demisto: [Administrator SecondShift]

User Invitations

Before inviting users to your Cortex XSOAR environment, you need to add an email integration instance, such as EWS, Gmail, EWS, Mail Sender, and so on.

After adding an email integration instance, you can send emails to a user's email address. You must add their role and choose whether to invite them to a specific incident.

After you [Invite a User](#), an invitation is valid for one week from the time it was sent. If it is not accepted, the invitation expires, unless the invite expiration is resent. Once users accept the invite, they have access and permission within Cortex XSOAR, according to the roles you assign them.

If you need to connect to an external host name, add the **External Host Name** in the **Server Configuration** section in the **Troubleshooting** page, otherwise the URL invite may not be valid.

In the **Invites** tab, you can see details of all the sent invitations including the invite URL and whether they have been accepted.

You can also take the following actions:

Action	Description
Delete the invite	If the user has not accepted the invitation you can delete the invite and the user cannot use this invitation to join the Cortex XSOAR environment.
Resend the invite	The user receives another invitation. The expiration date extends for another week.
Reset the expiration	The expiration for the selected invitation is extended for one more week.

Invite a User

You can invite one user at a time and add their roles as required. The invitation is sent by email to the user's email address or you can copy the URL invite and send it direct. An invite is valid for one week.

Ensure that you configure an Email integration, such as Gmail, EWS, Mail Sender, and so on.

You may need to add an external host name if your Cortex XSOAR needs to connect to an external host name.

STEP 1 | Select **Settings > Users and Roles > Invites**.

STEP 2 | Click **Invite User**.

STEP 3 | Type the email address of the user you want to add.

STEP 4 | From the drop down list, select at least one role to assign to the user.

STEP 5 | (Optional) Select an incident to assign the user.

STEP 6 | Click **Invite**

Details of the invitation appears in the **Invites** tab.

The user is sent an email with details on how to join. Once accepted the user's details appear in the **Users** tab.

Integration Permissions

You can use Role Based Access Control (RBAC) to assign commands at the integration instance level. If you have multiple instances of the same integration, you can assign different roles (permission levels) for the same command in each instance.



To restrict access to integrations using RBAC, see [Roles in Cortex XSOAR](#).

If you want to view or edit integration permissions, go to **Settings > USERS AND ROLES > Integration Permissions**. You can see a list of all the enabled integrations in Cortex XSOAR. Under each integration you can see the following:

- **Commands:** a list of all commands for the integration.
- **Instance:** a list of all instances for the integration.
- **Permitted roles:** a drop down list of roles you can assign to the command.

Users that do not have command permissions cannot do the following:

- Run the command from the CLI (!).
- Complete pending tasks in a Work Plan that run the command.
- Edit arguments for playbook tasks that have this command.
- Select the command when editing a playbook.
- Select a script if using a reputation-command.

Password Policy

You can set a password policy for all internal users in Cortex XSOAR. The password policy enables you to do the following:

- Set password complexity requirements.
- Set the password expiry and prevent repetition passwords (remembers up to the last 30 passwords).
- Brute-force prevention (user lockout after a number of attempts).

By default, Cortex XSOAR provides the following default password policy:

- A FIPS compliant password policy in the **Password Policy** tab. To create a password policy, see [Create a Password Policy](#).
- New installations: when installing Cortex XSOAR in interactive mode, you must have a strong password. If installing with the `-y` option you can use a non FIPS compliant password, but you must change the password upon first log in.
- Upgrade: all internal users with a non FIPS compliant password may need to change their password upon next log in. Password expiry and repetition prevention does not work retroactively and passwords before the upgrade are not considered. For existing passwords, the expiry countdown starts from the upgrade time.

Before installing or upgrading, you can change the password policy by adding a server configuration, as described in [Edit a Default Password Policy](#). However, as soon as you make password changes in the **Password Policy** tab, these settings override the server configuration settings. All further changes are made in the **Password Policy** tab.

Create a Password Policy

In Cortex XSOAR you can set a default FIPS compliant password policy in the **Password Policy** tab. Any changes in the **Password Policy** override any password changes made in the server configuration settings.

STEP 1 | Go to **Settings > USERS AND ROLES > Password Policy**.

STEP 2 | In the **Enable Password Policy** section, select **On**.

STEP 3 | Add the password requirements, as necessary.

The 0 value disables the settings.

STEP 4 | When selecting **unlock** choose one of the following options to unlock the user's account:

- **By Admin only:** only administrators can manually unlock user accounts.
- **Automatically:** users can unlock themselves after a specified time.

Locked out users cannot use API keys. Cortex XSOAR has a delay mechanism for multiple failed logins. However, unlike the lockout mechanism, this system is not suitable for preventing automated brute-force attacks. It is useful for preventing accidental lockouts.

STEP 5 | Click **Save**.

Edit a Default Password Policy

When installing or upgrading Cortex XSOAR, users with non FIPS compliant passwords may need to change their password upon next log in. You can change this behavior before upgrading or installing by adding a server configuration.



After adding a server configuration, you can see the changes in **Settings > USERS AND ROLES > Password Policy**. However, when you make any changes in the **Password Policy** tab, these override the changes made in the **Server configuration**. You then make any future changes in the **Password Policy** tab.

STEP 1 | Go to **Settings > About > Troubleshooting**.

STEP 2 | In the **Server Configuration** section, click **Add Server Configuration**.

STEP 3 | Add the keys and values, as described in [Default Password Policy Keys](#).

STEP 4 | Click **Save**.

Instead of adding the keys separately, you can enter one key containing all of the information in the following format:

```
{
  "Server": {
    "HttpsPort": "443"
  },
  "db": {
    "index": {
      "entry": {
        "disable": true
      }
    }
  },
  "limit": {
    "docker": {
      "memory": true
    }
  },
  "password": {
    "policy": {
      "default": {
        "Enabled": true,
        "MinLowercaseChars": 4,
        "MinUppercaseChars": 4,
        "ExpireAfter": 4,
        "ExpireUnit": "day",
        "PreventRepetition": true,
        "MaxFailedLoginAttempts": 4,
        "SelfUnlockAfterMinutes": 4
      }
    }
  }
}
```

Default Password Policy Keys

By default, when installing or upgrading, users may need to change their passwords. You can change this behavior by configuring a server, as described in [Edit a Default Password Policy](#) and add the following keys and values.

Keys	Default values
<code>password.policy.default.enabled</code>	Default is true.

Keys	Default values
MinPasswordLength	Default is 8.
MinLowercaseChars	Default is 1.
MinUppercaseChars	Default is 1.
MinDigitsOrSymbols	Default is 1.
ExpireAfter	Default is 0. Numeric settings are disabled with a 0 value.
ExpireUnit	Values are month, week and day. Default is Month.
PreventRepetition	Default is true.
MaxFailedLoginAttempts	Default is 10. How many attempted log ins within one minute. For example, 10 failed logins within one minute.
SetUnlockAfterMinutes	Default is 0. User unlocks automatically after x minutes (if 0, only an administrator can unlock).

Change the Administrator Password

Use this procedure when the administrator cannot log in and has forgotten the password. To change a password, you need to create a new administrator and then you can change the password for the current administrator.

You create the new administrator by creating a one-time configuration (OTC) file, in which you define the user configurations. After the file is saved, restart the Cortex XSOAR server. The OTC file is automatically deleted.

STEP 1 | Define a new administrator, by creating a `/var/lib/demisto/otc.conf.json` file with content similar to the following.

Ensure the file has `demisto:demisto` ownership.

```
{
  "users": [{
    "username": "newadmin",
    "password": "veryStrongPassword!",
    "email": "admin@company.com",
    "phone": "+650-123456",
    "name": "New Admin Dude",
    "roles": {
      "demisto": [
        "Administrator"
      ]
    }
  }]
}
```

STEP 2 | Save the file and restart the Cortex XSOAR server by running the `systemctl restart demisto` command.

The file is removed when Cortex XSOAR restarts.

STEP 3 | Log in to Cortex XSOAR by using the new administrator credentials, as created in step 1.

STEP 4 | Change the password for the current administrator and log out.

STEP 5 | Login to Cortex XSOAR using the current administrator credentials, including the new password.

STEP 6 | Remove the new administrator you created in step 1.

Authenticate Users with SAML 2.0

SAML exchanges authentication and authorization data between security domains. SAML 2.0 is an XML-based protocol that uses security tokens containing assertions to pass information about a principal (usually an end user) between a SAML authority (Identity Provider) and a SAML consumer (Service Provider).

SAML 2.0 enables web-based authentication and authorization scenarios including cross-domain single sign-on (SSO), which helps reduce the administrative overhead of distributing multiple authentication tokens to the user. For more information about SAML 2.0, see [SAML 2.0 Wikipedia](#).

You can authenticate your Cortex XSOAR users using SAML 2.0 authentication with your identity provider, such as Okta. You need to define Cortex XSOAR authentication in your Identity Provider's account, then create a SAML 2.0 instance in Cortex XSOAR.

Set up Okta as the Identity Provider Using SAML 2.0

You can authenticate your Cortex XSOAR users using SAML 2.0 authentication and Okta as the identity provider. You need to authenticate Cortex XSOAR in your Okta account, and create a SAML 2.0 instance in Cortex XSOAR by completing the following procedures:

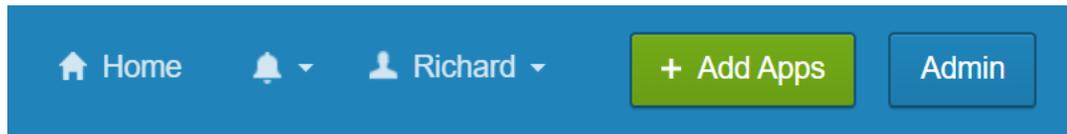
- [Create Okta Groups for Cortex XSOAR Users](#)
- [Define the Okta Application to authenticate Cortex XSOAR](#)
- [Configure the SAML 2.0 Integration for Okta](#)
- [Map Okta Groups to Cortex XSOAR Roles](#)

Create Okta Groups for Cortex XSOAR Users

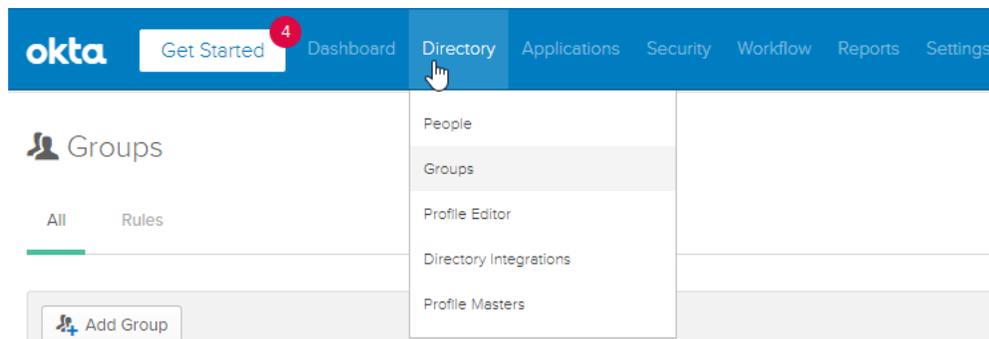
To authenticate Cortex XSOAR users with Okta, you need to have at least one Okta group that defines Cortex XSOAR users, which eventually is mapped to Okta roles. There are two common methods for grouping and mapping users:

- Create a single Okta group for all users. For example, Cortex All Users.
- Create an Okta group for each business unit. For example, Cortex XSOAR IT, Cortex XSOAR Analysts, Cortex XSOAR Admins.

STEP 1 | Log in to Okta and click **Admin**.



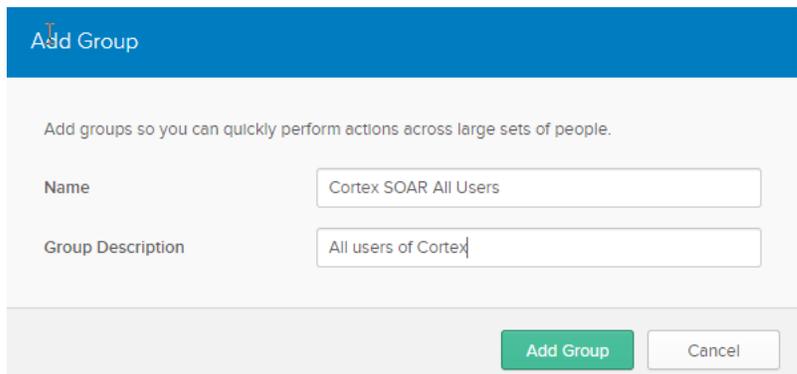
STEP 2 | Log in to Okta and select **Directory > Groups**.



STEP 3 | Click **Add Group**.

STEP 4 | Type a name and description for the group.

The name and description should enable you to easily identify the users of that group.



STEP 5 | To add users to the group, select the group name you created in step 4.

STEP 6 | Click **Manage People** and add the users as required.

STEP 7 | Click **Save**.

STEP 8 | Define the Okta Application to authenticate Cortex XSOAR.

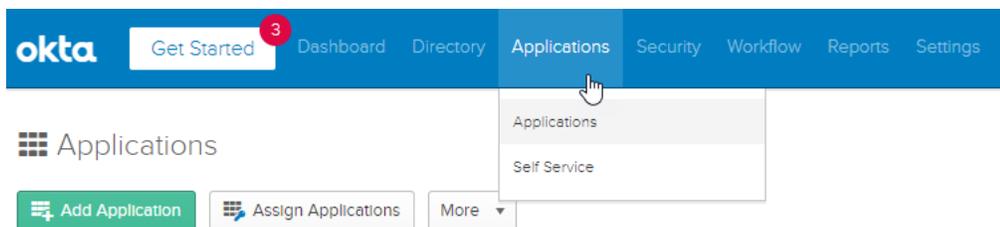
Define the Okta Application to authenticate Cortex XSOAR

You need to define the Okta application to use it in Cortex XSOAR.

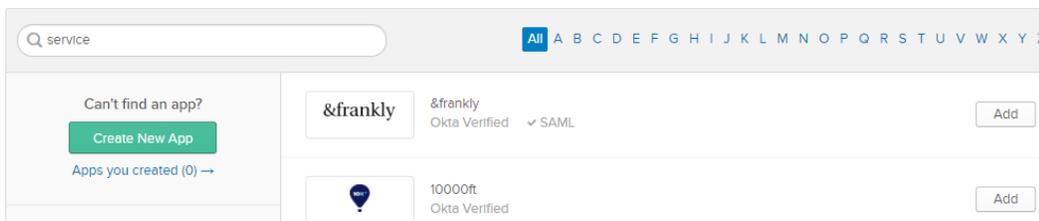
Before you start you need to create a group as described in [Create Okta Groups for Cortex XSOAR Users](#).

STEP 1 | Log in to Okta and click **Admin**.

STEP 2 | Select **Applications > Applications > Add Application**.



STEP 3 | Click **Create New App**.



STEP 4 | From the **Create a New Application Integration** window, in the **Platform** field, select **Web**.

STEP 5 | In the **Sign on method** field, select **SAML 2.0**.

The screenshot shows a dialog box titled "Create a New Application Integration" with a close button (X) in the top right corner. The dialog contains the following elements:

- Platform:** A dropdown menu currently showing "Web".
- Sign on method:** Three radio button options:
 - Secure Web Authentication (SWA)
Uses credentials to sign in. This integration works with most apps.
 - SAML 2.0
Uses the SAML protocol to log users into the app. This is a better option than SWA, if the app supports it.
 - OpenID Connect
Uses the OpenID Connect protocol to log users into an app you've built.
- Buttons:** A green "Create" button and a white "Cancel" button at the bottom right.

STEP 6 | Click **Create**.

STEP 7 | From the **General Settings** section, in the **App name** field, type a name for the application and click **Next**.

STEP 8 | In the **SAML Settings** section, add the following parameters:

- The **GENERAL** parameters
- The **ATTRIBUTE STATEMENTS** parameters.
- The **GROUP ATTRIBUTE STATEMENTS** parameters

For information about the parameters, see [SAML Settings for the Okta Application](#).

STEP 9 | Click **Next**.

STEP 10 | Select the following options:

3 Help Okta Support understand how you configured this application

Are you a customer or partner? I'm an Okta customer adding an internal app
 I'm a software vendor. I'd like to integrate my app with Okta

 The optional questions below assist Okta Support in understanding your app integration.

App type  This is an internal app that we have created

STEP 11 | Click **Finish**.

When setting up the integration in Cortex XSOAR, you need to add the setup instructions and Identity Provider metadata.

 **SAML 2.0** is not configured until you complete the setup instructions.

[View Setup Instructions](#)

[Identity Provider metadata](#) is available if this application supports dynamic configuration.

STEP 12 | Continue with [Configure the SAML 2.0 Integration for Okta](#).

SAML Settings for the Okta Application

The following tables describes the SAML settings for Otka.

General Parameters

Parameter	Value
Single sign on URL	https://<cortexxsoarURL>/saml
Audience URI (SP Entity ID)	https://<cortexxsoarURL>/saml/metadata
Default RelayState	Keep this field empty.
Name ID format	EmailAddress. The Cortex XSOAR username is the user's email address, as defined in Okta.
Application username	Okta username.
Update application user name on	Create and update

A SAML Settings

GENERAL

Single sign on URL ?
 Use this for Recipient URL and Destination URL
 Allow this app to request other SSO URLs

Audience URI (SP Entity ID) ?

Default RelayState ?
If no value is set, a blank RelayState is sent

Name ID format ?

Application username ?

Update application username on

Attribute Statement Parameters

Name	Name Format	Value
FirstName	Unspecified	user.firstName
LastName	Unspecified	user.lastName
Email	Unspecified	user.email
login	Unspecified	user.login
Phone	Unspecified	user.primaryPhone

ATTRIBUTE STATEMENTS (OPTIONAL) [LEARN MORE](#)

Name	Name format (optional)	Value
FirstName	Unspecified	user.firstName
LastName	Unspecified	user.lastName
Email	Unspecified	user.email
login	Unspecified	user.login
Phone	Unspecified	user.primaryPhone

Add Another

Group Attribute Statement parameters

Name	Name format	Filter	Name of group
memberof	Unspecified	Equals	Name of the group you have created.

The Group Attribute Statement parameters define which groups to associate with Cortex XSOAR and which groups are to be mapped to Cortex XSOAR roles. In this example, add a group called Everyone.

GROUP ATTRIBUTE STATEMENTS (OPTIONAL)

Name	Name format (optional)	Filter
memberof	Unspecified	Equals

Add Another

 If you are using **memberof** as a group attribute statement, ensure not to use the **memberof** as an attribute statement. You cannot have both single user and group user attributes.

Configure the SAML 2.0 Integration for Okta

You need to configure the SAML 2.0 integration so you can use it in Cortex XSOAR.

Before you start, access the Okta Setup Instructions and Identity Provider metadata in Okta, as described in [Set up Okta as the Identity Provider Using SAML 2.0](#).

STEP 1 | Go to **Settings > Integrations > Servers & Services**.

STEP 2 | Search for **SAML 2.0**

STEP 3 | Click **Add instance** to configure a new integration.

STEP 4 | Add the [SAML 2.0 Okta Parameters](#).

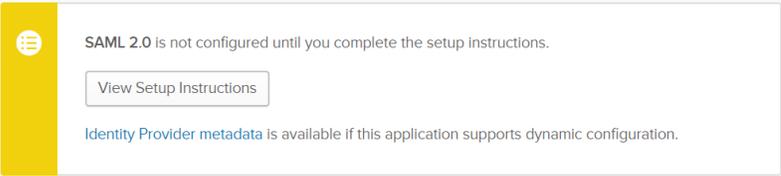
STEP 5 | Click **Test** to validate the URLs, token, and connection.

STEP 6 | To verify that the settings are successful, in the instance settings, click **Get service provider metadata**.

STEP 7 | [Map Okta Groups to Cortex XSOAR Roles](#).

SAML 2.0 Okta Parameters

The following table describes the SAML 2.0 parameters for Okta, when adding a new instance in Cortex XSOAR:

Attribute	Description
Name	A name for the integration instance.
Service Provider Entity ID	The URL of your Cortex XSOAR server (also known as an ACS URL). In the format: <code>https://yourdomain.com/saml</code>
Idp metadata URL	URL of your organization's IdP metadata file. You can find this in the Sign On tab in Okta or when defining an Okta application, as described in Define the Okta Application to authenticate Cortex XSOAR . 
IdP metadata file	Your organization's IdP metadata file. You either need to add the IdP metadata URL or the file.
IdP SSO URL	The URL of the IdP application that corresponds to Cortex XSOAR. You can copy and paste the IdP SSO URL in Okta, when clicking View Setup Instructions .
Attribute to get username	Attribute in your IdP for the user name.
Attribute to get email	Attribute in your IdP for the user's email address.
Attribute to get first name	Attribute in your IdP for the user's first name.
Attribute to get last name	Attribute in your IdP for the user's last name.
Attribute to get phone	Attribute in your IdP for the user's phone number.
Attribute to get groups	Attribute in your IdP for the groups of which the user is a member.
Groups delimiter	Groups list separator.
Default role	Role to assign to the user when they are not a member of any group.

Attribute	Description
RelayState	Only used by certain IdPs. If your IdP uses relay state, you need to supply the relay state.
Sign request and verify response signature	Method for the IdP to verify the user sign-in request using the IdP vendor certificate.
Identity Provider public certificate	Public certificate for your IdP.
Identity Provider private key	Private key for your IdP, in PEM format. Created locally by the user who wants to use SAML. The matching public key is uploaded to Okta.
Do not map SAML groups to Cortex XSOAR roles	SAML groups will not be mapped to Cortex XSOAR roles.

Map Okta Groups to Cortex XSOAR Roles

It is important that when you specify the Okta group in Cortex XSOAR to map to a role that you use the exact group name as it appears in Okta. Alternatively, you can specify `*`, which will pass all Okta groups to the relevant Cortex XSOAR roles (this is not recommended).

STEP 1 | Go to **Settings > Users & Roles > Roles**.

STEP 2 | Create or edit an existing role, as described in [Define a Role](#).

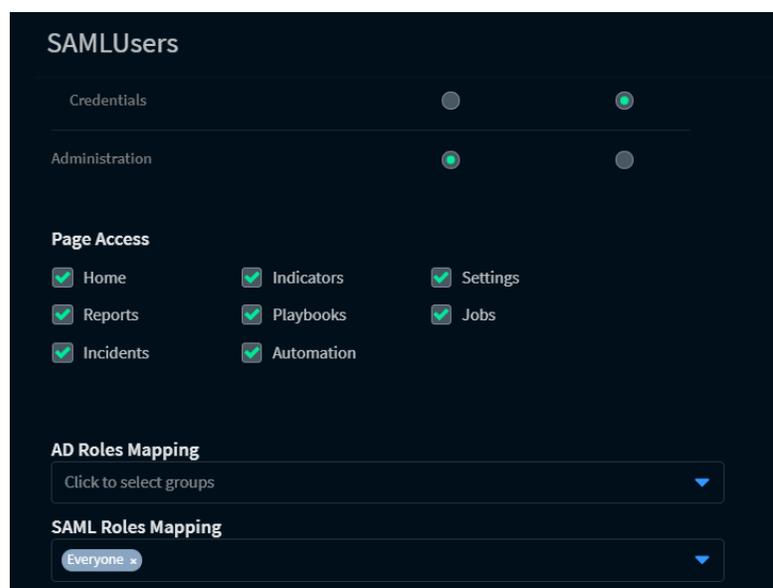
STEP 3 | In the **SAML Roles Mapping** field, specify one or more SAML groups to map to the Cortex XSOAR role.

In the following example, you want to add the group, called “Everyone”, which has been defined in the **Group Attributes Statement** field in Okta:

GROUP ATTRIBUTE STATEMENTS (OPTIONAL)

Name	Name format (optional)	Filter
memberof	Unspecified ▼	Equals ▼ Everyone
<input type="button" value="Add Another"/>		

In Cortex XSOAR, in the **SAML Roles Mapping** field, add the following:



Set up Microsoft Azure as the Identity Provider

You can authenticate your Cortex XSOAR users using SAML 2.0 authentication and Microsoft Azure (Azure) as the identity provider. You need to authenticate Cortex XSOAR in your Azure account, and create a SAML 2.0 instance in Cortex XSOAR, by completing the following procedures:

- [Create a Non-Gallery Application in Azure](#)
- [Define Azure to authenticate Cortex XSOAR](#)
- [Configure the SAML 2.0 Integration for Azure](#)
- [Map Azure Groups to Cortex XSOAR Roles](#)

Troubleshooting (generic - known errors)

The following are known issues when using Single sign on in Azure:

- **Method Not Allowed:** Ensure the endpoint is used for the Service Provider Entity ID and Reply URL for the IdP and Service provider, in the format: `https://demisto-dns/saml`.
- `{"id":"errSAMLLogin","status":400,"title":"Failed to login via SAML","detail":"Failed to login via SAML","error":"","encrypted":false,"multires":null}"`: Most likely an attribute mapping issue. Ensure that all attributes that appear in Cortex XSOAR SAML 2.0 configuration are reflected in Azure claims and its associated SAML assertion. Attributes are case sensitive.

You may also receive this message, if you select the **Don't map SAML groups to Demisto Roles** checkbox and you do not define a role in **Default role (for IdP users without groups)** in the SAML2.0 configuration.

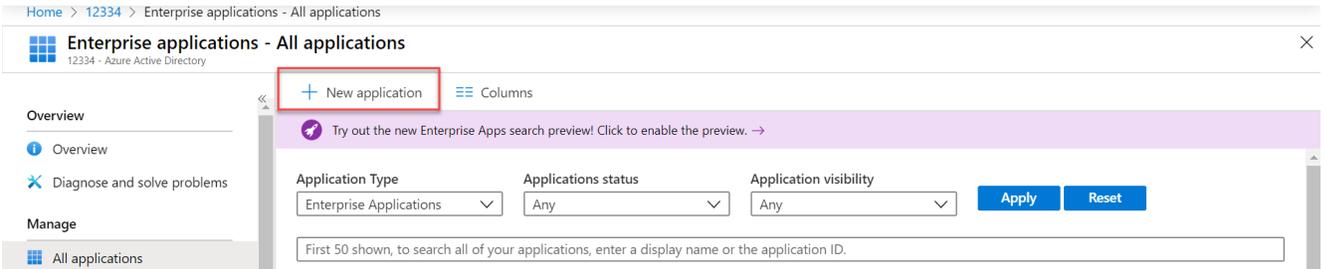
- After connecting through SSO, a user may temporarily see the home screen, but immediately returns to the login page. The user does not have any group assigned, so he cannot login.

Check the group mapping and see whether the `memberOf` attribute is correct. As a workaround, if you did not set the group mapping, you can use the **Default role (for IdP users without groups)** in the SAML2.0 configuration.

Create a Non-Gallery Application in Azure

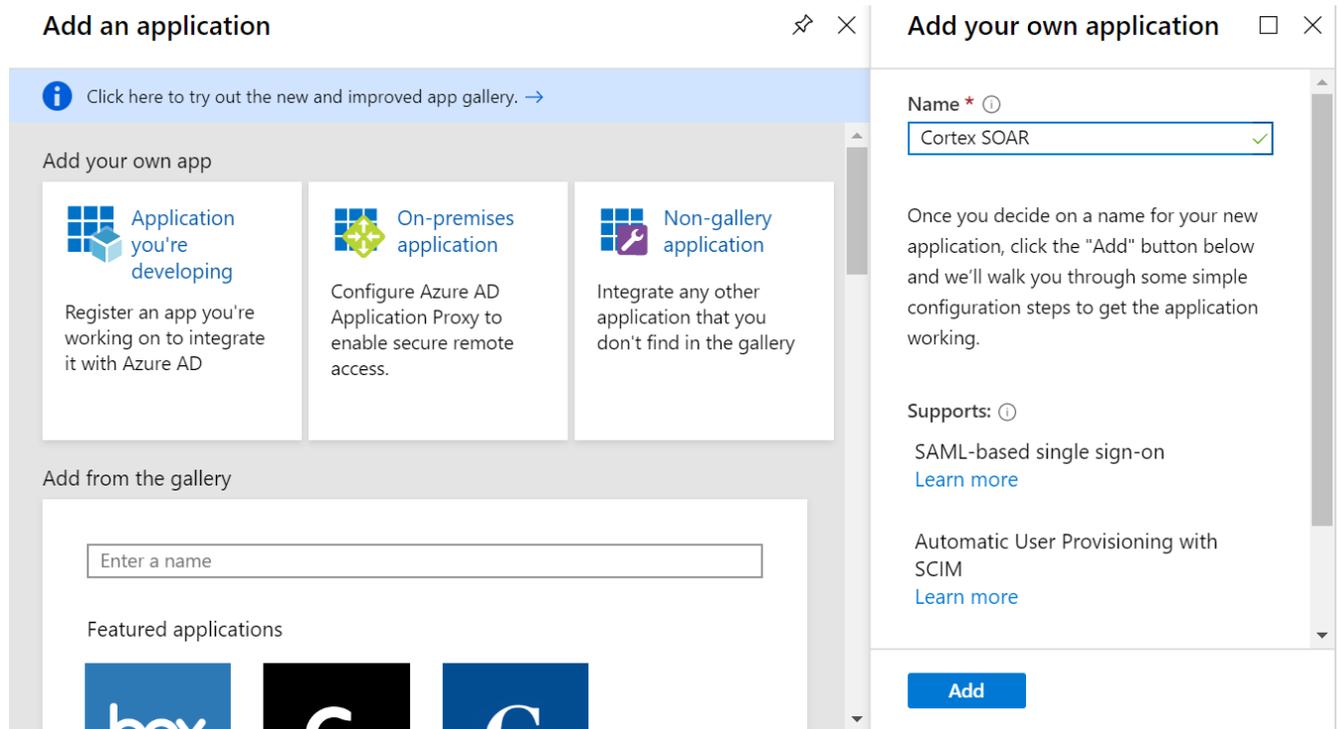
In Azure you need to create a non-gallery application and add groups to your application, before configuring single sign-on. Once completed, you can then configure the SAML 2.0 instance using Azure in Cortex XSOAR.

STEP 1 | From the home page, select **Azure Active Directory > Enterprise applications > New Application**.



STEP 2 | Select **Non-gallery application**.

STEP 3 | Type the name of your application and click **Add**.



STEP 4 | If you have not created any users or groups, go to **Home > Azure Active Directory** and select the following:

- For users, select **Users > New user** and create or invite a user.
- For groups, select **Groups > New Group** and type the group information as required.

STEP 5 | From the **Name of the Application Overview** window, in the **Getting Started** section, click **Assign users and groups**.

Getting Started

**1. Assign users and groups**

Provide specific users and groups access to the applications

[Assign users and groups](#)

To access the Application overview window, go to **Home > Azure Active Directory > Enterprise applications** and then select your application, created in step 3.

STEP 6 | In the ***Name of the Application* Users and groups** window, click **Add user**.

STEP 7 | In the **Add Assignment** window, click **Users and groups**.

STEP 8 | Select the group/users as required and click **Select**.

STEP 9 | Continue with [Define Azure to authenticate Cortex XSOAR](#).

Define Azure to authenticate Cortex XSOAR

You need to authenticate Cortex XSOAR for single sign-on section in your Azure environment. Before you start, ensure that you have created a non-gallery application and assigned users, as described in [Create a Non-Gallery Application in Azure](#).

STEP 1 | Go to **Home > Enterprise applications > Name of your application**.

STEP 2 | In the **Set up single sign on** section, click **Get started**.

**2. Set up single sign on**

Enable users to sign into their application using their Azure AD credentials

[Get started](#)

STEP 3 | Click **SAML**.

STEP 4 | In the **Basic SAML Configuration** section, add the **Identifier (Entity ID)** and **Reply URL (Assertion Consumer Service URL)**.

1 Basic SAML Configuration 

Identifier (Entity ID)	https://eu-west-1.compute.amazonaws.com/saml
Reply URL (Assertion Consumer Service URL)	https://eu-west-1.compute.amazonaws.com/saml
Sign on URL	Optional
Relay State	Optional
Logout Url	Optional

STEP 5 | In the **User Attributes & Claims** section, click the edit icon and add the following attributes and values as required.

User Attributes & Claims	
Email	user.mail
FirstName	user.givenname
LastName	user.surname
Phone	"123456789"
Unique User Identifier	user.userprincipalname

Ensure the attribute names match the names in Cortex XSOAR, when defining the instance.

STEP 6 | To add a new group, click **Add a group claim**.

STEP 7 | In the **Group Claims (Preview)** window, select **Security groups**.

STEP 8 | In the **Advanced options** section, select the **Customize the name of the group claim** check box.

STEP 9 | In the **Name** field, type **memberOf**.

Group Claims (Preview)

Manage the group claims used by Azure AD to populate SAML tokens issued to your app

Which groups associated with the user should be returned in the claim?

None
 All groups
 Security groups
 Directory roles

Source attribute *

Group ID

Advanced options

Customize the name of the group claim

Name (required)

memberOf

Save

STEP 10 | Click **Save**.

Ensure that you have a group assigned to a user in the Cortex XSOAR in Azure. The Object ID is added to the **SAML Roles Mapping** field in Cortex XSOAR.

CA

Cortex Admins

Admin Permissions

Membership type	Assigned
Source	Cloud
Type	Security
Object Id	39124842-483e-4ffa-93ed-5f18367c77bc
Creation date	1/8/2020, 3:11:13 PM

Direct members

3 User(s)
 0 Group(s)
 0 Device(s)
 0 Other(s)

Group memberships

0
 0

STEP 11 | Note the **App Federation Metadata Url** and **Login URL**, which are needed to configure the instance in Cortex XSOAR.

3
SAML Signing Certificate ✎

Status	Active
Thumbprint	1D0C6E55AAEF9A370C74EA592C02D1E97C1617F4
Expiration	1/8/2023, 3:15:42 PM
Notification Email	anyname@name.com
App Federation Metadata Url	https://login.microsoftonline.com/934a6d32-95... 📄
Certificate (Base64)	Download
Certificate (Raw)	Download
Federation Metadata XML	Download

4
Set up Cortex SOAR

You'll need to configure the application to link with Azure AD.

Login URL	https://login.microsoftonline.com/934a6d32-95... 📄
Azure AD Identifier	https://sts.windows.net/934a6d32-9550-4fd2-b... 📄
Logout URL	https://login.microsoftonline.com/common/wsf... 📄

[View step-by-step instructions](#)

STEP 12 | You can now add an instance in Cortex XSOAR, as described in [Configure the SAML 2.0 Integration for Azure](#).

Configure the SAML 2.0 Integration for Azure

Before you configure an instance of the SAML 2.0 integration in Cortex XSOAR, you need to configure Azure. For more information, see [Set up Microsoft Azure as the Identity Provider](#).

STEP 1 | Go to **Settings > Integrations > Servers & Services**.

STEP 2 | Search for **SAML 2.0**

STEP 3 | Click **Add instance** to configure a new integration.

STEP 4 | Add the parameters, as described in [SAML 2.0 Azure Parameters](#).

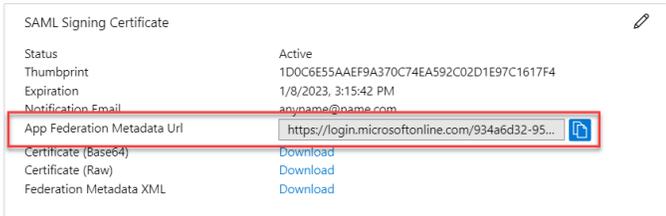
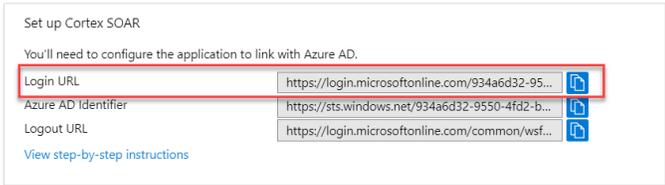
STEP 5 | Click **Test** to validate the URLs, token, and connection.

STEP 6 | To verify that the settings are successful, in the instance settings, click **Get service provider metadata**.

STEP 7 | To map Azure groups, continue with [Map Azure Groups to Cortex XSOAR Roles](#).

SAML 2.0 Azure Parameters

The following table describes the SAML 2.0 parameters for Azure, when adding a new instance in Cortex XSOAR:

Attribute	Description
Name	A name for the integration instance.
Service Provider Entity ID	The URL of your Cortex XSOAR server (also known as an ACS URL). In the format: https://yourdomain.com/saml
Idp metadata URL	URL of your organization's IDP metadata file. You can copy this from the App Federation Metadata URL in the SAML Signing Certificate in Azure. 
IdP metadata file	Your organization's IdP metadata file. You either need to add the Idp metadata URL or the file.
IdP SSO URL	The URL of the IdP application that corresponds to Cortex XSOAR. You can copy this from the Login URL field in the SAML Signing Certificate section. 
Attribute to get username	Attribute in your IdP for the user name. Value: nameIdentifier
Attribute to get email	Attribute in your IdP for the user's email address. Value: Email
Attribute to get first name	Attribute in your IdP for the user's first name. Value: FirstName

Attribute	Description
Attribute to get last name	Attribute in your IdP for the user's last name. Value: LastName
Attribute to get phone	Attribute in your IdP for the user's phone number. Value: Phone
Attribute to get groups	Attribute in your IdP for the groups of which the user is a member. Value: memberOf
Groups delimiter	Groups list separator. Value: " , "
Default role (for IdP users without groups)	Role to assign to the user when they are not a member of any group. For example, Analyst .
RelayState	Only used by certain IdPs. If your IdP uses relay state, you need to supply the relay state.
Use system proxy settings	Select the check box to use proxy settings.
Compress encode URL (AFDS)	(Mandatory) Select the check box to compress encode URL (AFDS). If not, you may receive a Decoding Flat error during connection.
Service identifier (AFDS)	Add the appid value, which can be found at the end of the IDP metadata URL. For example, https://login.microsoftonline.com/934a6d32-9550be/federationmetadata/2007-06/federationmetadata.xml?appid=b0331331-f15b-4a32-9f48-19158beb0340 .
Do not map SAML groups to Cortex XSOAR roles	SAML groups are not mapped to Cortex XSOAR roles. Default roles are assigned and you can select them later.

Map Azure Groups to Cortex XSOAR Roles

It is important that when you add the SAML role in Cortex XSOAR to map it to the **Object ID** in Azure.

STEP 1 | Go to **Settings > Users & Roles > Roles**

STEP 2 | Create or edit an existing role, as described in [Define a Role](#).

STEP 3 | In the **SAML Roles Mapping** field, type the **Object ID** that appears in Azure.

For example, in Azure, we created a group, called Cortex Admins. Note the Object ID below:

[Delete](#) | [Got feedback?](#)

[Try out the new Groups experience improvements \(improved search and filtering\). Click to enable the preview. →](#)

CA

Cortex Admins

Admin Permissions

Membership type	Assigned 📄
Source	Cloud 📄
Type	Security 📄
Object Id	39124842-483e-4ffa-93ed-5f18367c77bc 📄
Creation date	1/8/2020, 3:11:13 PM 📄

Add the Object ID to the **SAML Roles Mapping** field in Cortex XSOAR:

Administrator

Playbooks	<input type="radio"/>	<input checked="" type="radio"/>
Settings	<input type="radio"/>	<input type="radio"/>
Users	<input type="radio"/>	<input checked="" type="radio"/>
Integrations	<input type="radio"/>	<input checked="" type="radio"/>
Credentials	<input type="radio"/>	<input checked="" type="radio"/>
Administration	<input type="radio"/>	<input checked="" type="radio"/>

Page Access

<input checked="" type="checkbox"/> Home	<input checked="" type="checkbox"/> Indicators	<input checked="" type="checkbox"/> Settings
<input checked="" type="checkbox"/> Reports	<input checked="" type="checkbox"/> Playbooks	<input checked="" type="checkbox"/> Jobs
<input checked="" type="checkbox"/> Incidents	<input checked="" type="checkbox"/> Automation	

AD Roles Mapping

Click to select groups ▼

SAML Roles Mapping

39124842-483e-4ffa-93ed-5f18367c77bc ▼

Nested Roles

▼

STEP 4 | Click **Save**.

Set up ADFS as the Identity Provider Using SAML 2.0

You can authenticate your Cortex XSOAR users using SAML 2.0 authentication and Active Directory Federation Services (ADFS) as the identity provider. You need to authenticate Cortex XSOAR in your ADFS account, and create a SAML 2.0 instance in Cortex XSOAR by completing the following procedures:

- [Create Relying Party Trust in ADFS](#)
- [Define the Claim Issuance Policy](#)
- [Configure the SAML 2.0 Integration for ADFS](#)
- [Map ADFS Groups to Cortex XSOAR Roles](#)

For troubleshooting issues such as DNS configuration, Endpoints and Idp initiated sign on, see [Microsoft ADFS Troubleshooting](#).

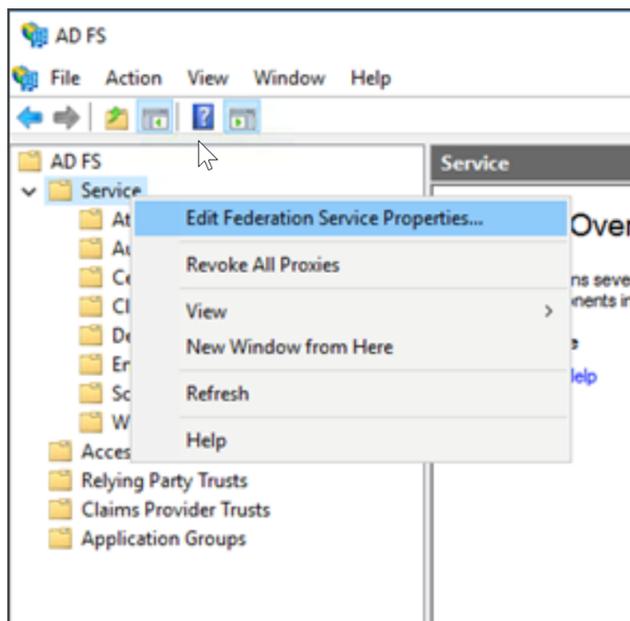
Create Relying Party Trust in ADFS

In ADFS you need to create a Relying Party Trust. The following procedure uses ADFS 3.0 on Windows Server 2016 and shows demistodev.local as the ADFS portal service which will allow a trust connection from the <https://demo.demisto.com> web server.

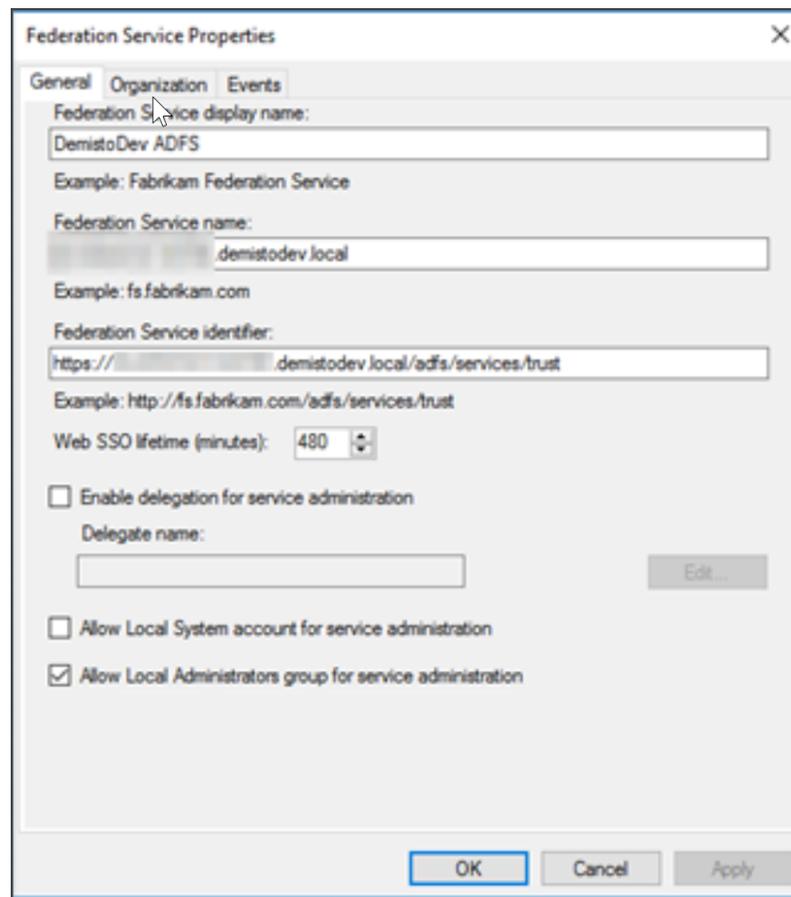
You must have a valid and trusted server certificate for ADFS to work, not the self-signed certificates that come with Cortex XSOAR. If you do not use a trusted server certificate for ADFS, you will experience TLS connection issues with ADFS and the integration will not work properly.

STEP 1 | Log in to the ADFS server management console.

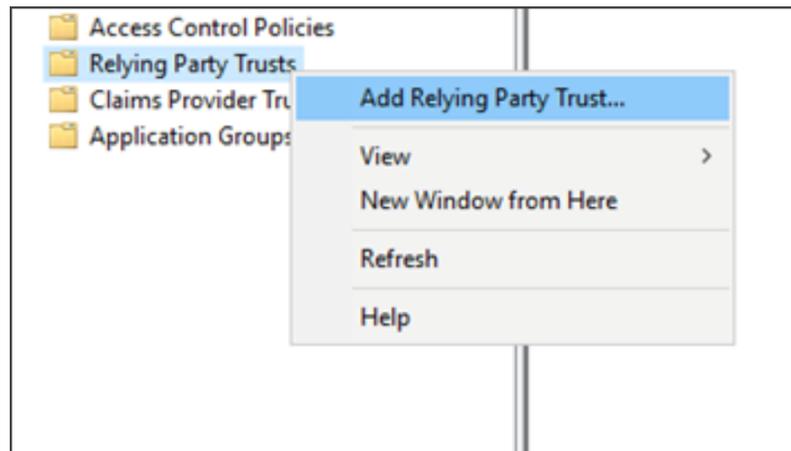
STEP 2 | In the tree in the left panel, right-click **Service** and select **Edit Federation Service Properties**.



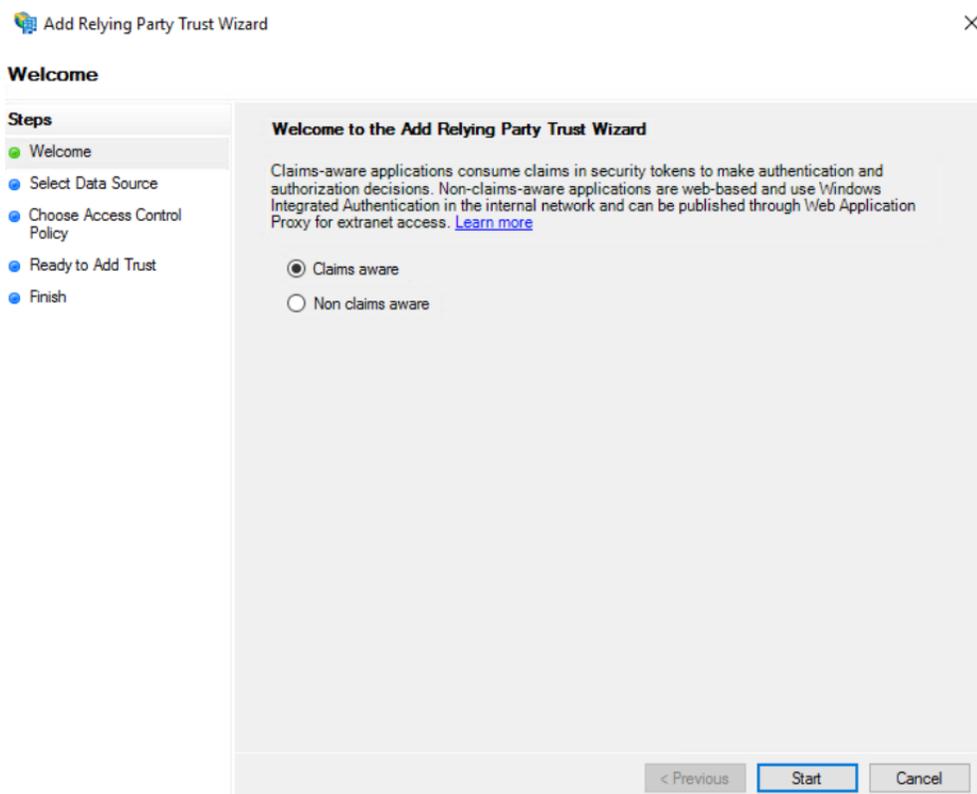
STEP 3 | Click the **General** tab and confirm that the DNS entries and certificates names are correct.



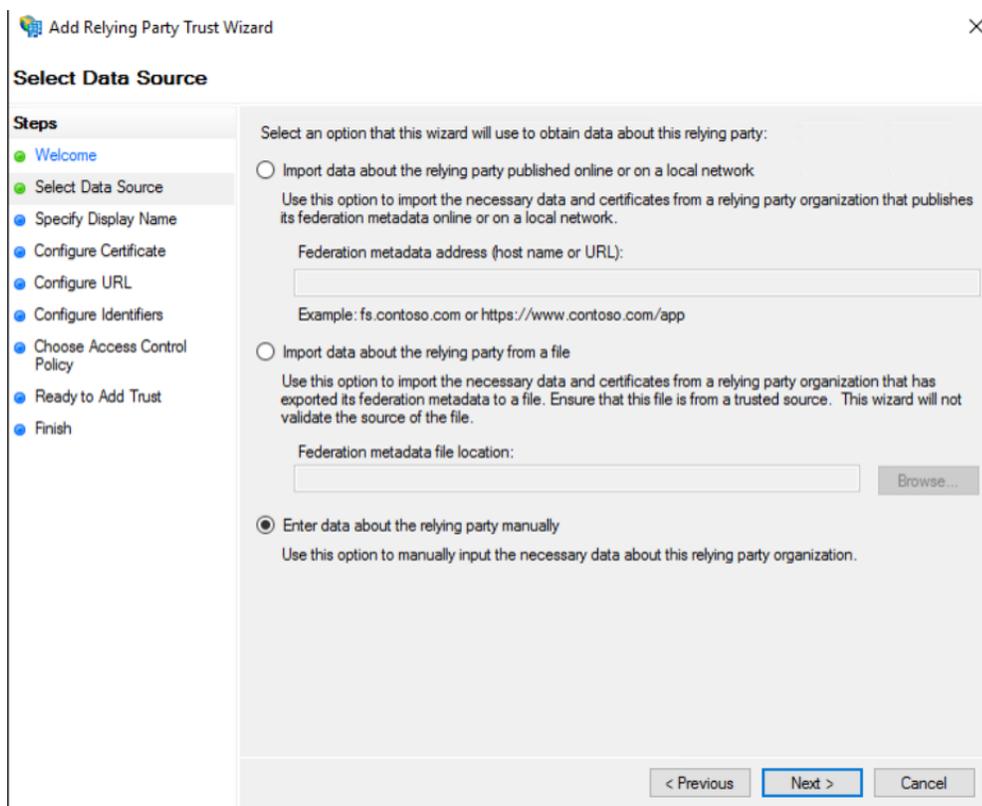
STEP 4 | In the tree in the left panel, right-click **Relying Party Trusts** and select **Add Relying Party Trusts**.



STEP 5 | The Add Relying Party Trust Wizard screen appears. Click **Start**.

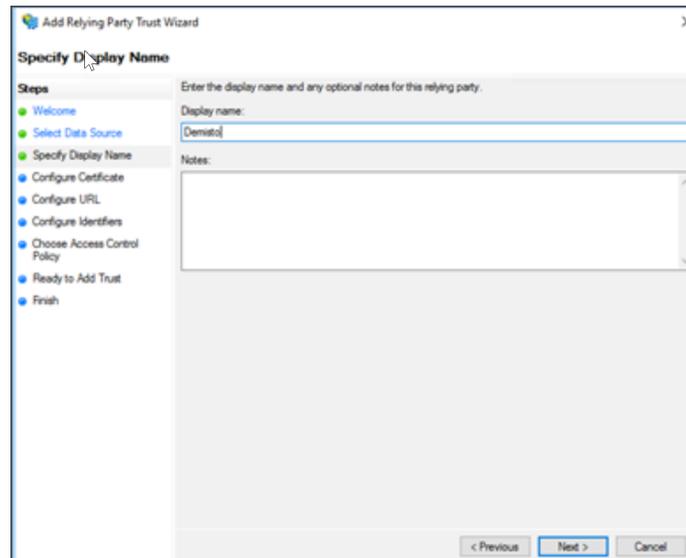


STEP 6 | In the Select Data Source page, select **Enter data about the relying party manually**.



STEP 7 | Click **Next**.

STEP 8 | In the Specify Display Name page, type a display name for the trust in the Display name field. In this example, the name of the trust is Demisto.

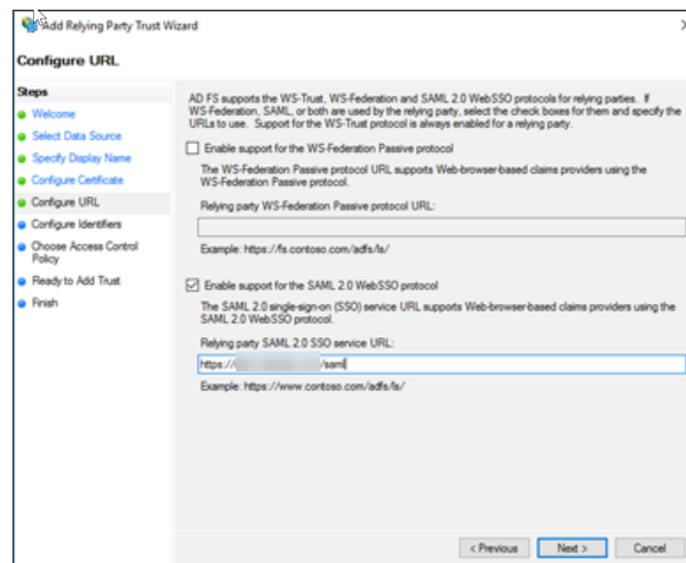


STEP 9 | Click **Next**.

STEP 10 | (Optional) In the Configure Certificate page, you can configure the claims encryption.

STEP 11 | Click **Next**.

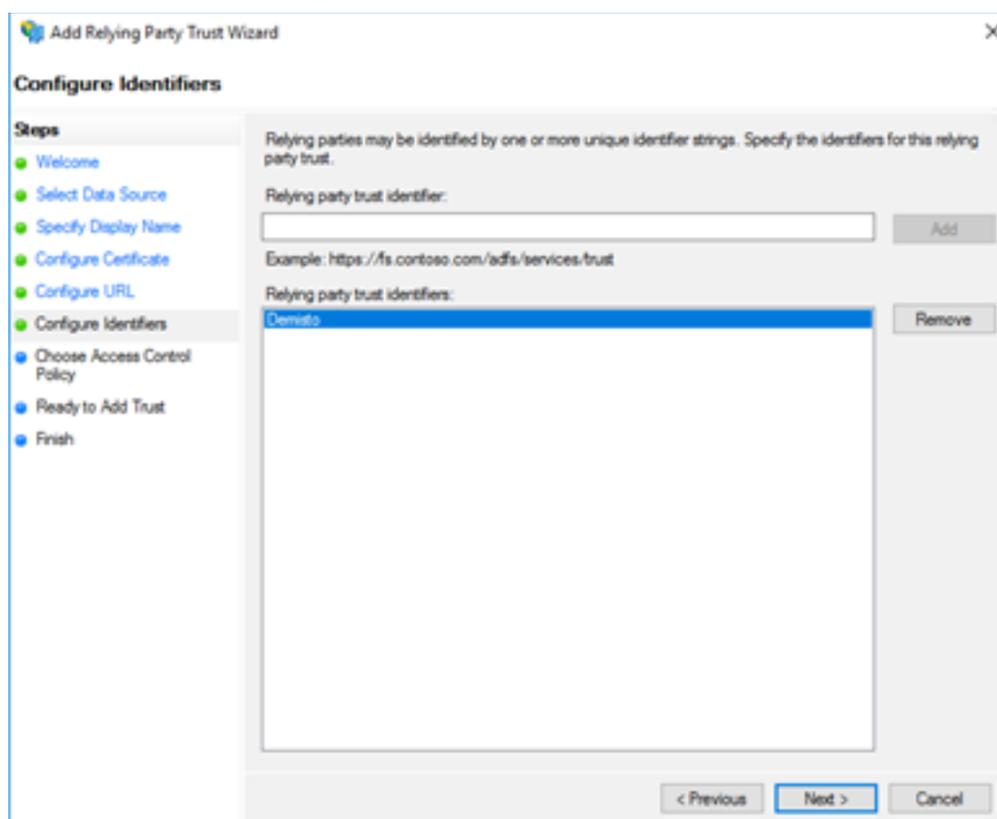
STEP 12 | In the Configure URL page, select **Enable support for the SAML 2.0 Web SSO protocol**, and enter the Cortex XSOAR server URL followed by /SAML.



STEP 13 | Click **Next**.

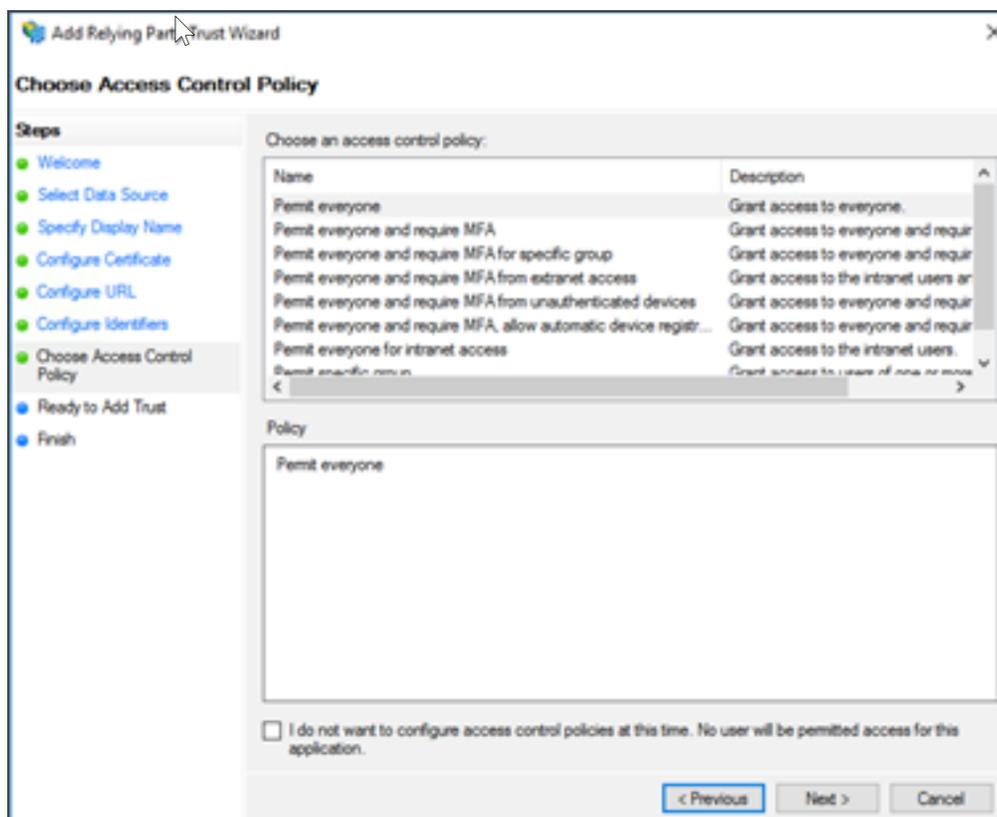
STEP 14 | In the Configure Identifiers page, add the Relying party trust identifier. The identifier can be a friendly name, the same as the Display name, or the application URL. This identifier is used to

redirect the user back to the Cortex XSOAR web server instead of asking the user to manually choose which service should log in to the ADFS IDP portal.



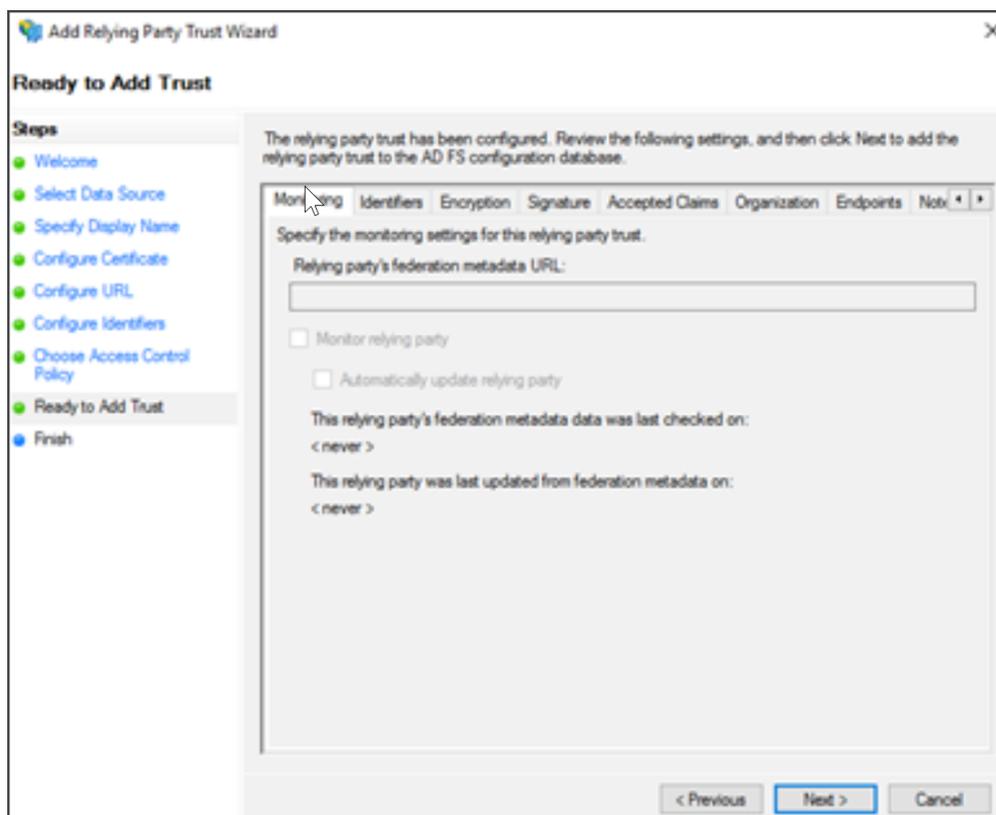
STEP 15 | Click **Next**.

STEP 16 | In the Choose Access Control Policy page, select an access control policy for the authentication portal. In this example, we choose .



STEP 17 | Click Next.

STEP 18 | In the Ready to Add Trust page, verify that all the settings are correct.



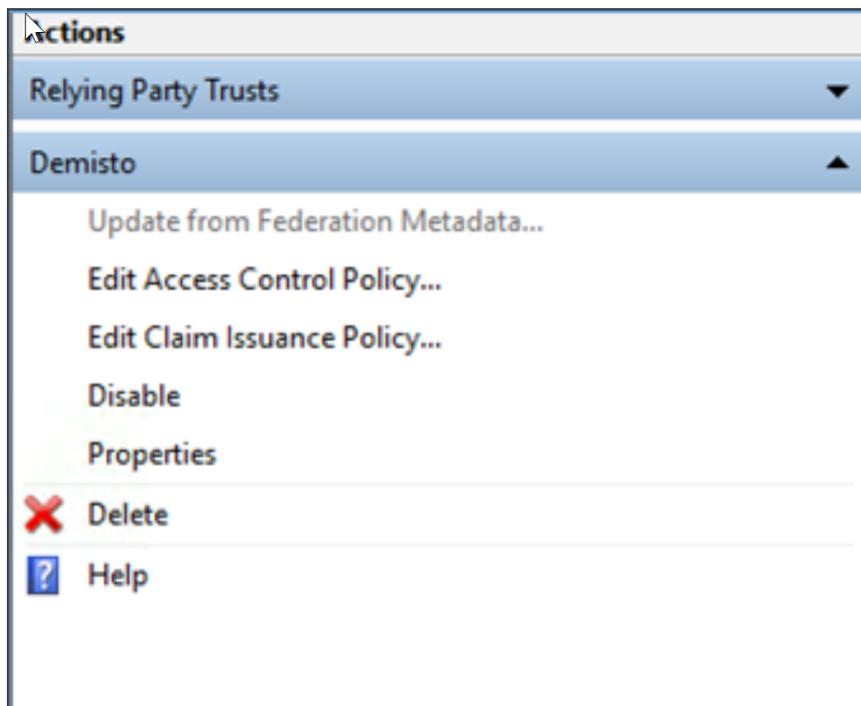
STEP 19 | Click **Next** and then click **Close**.

STEP 20 | Define the Claim Issuance Policy.

Define the Claim Issuance Policy

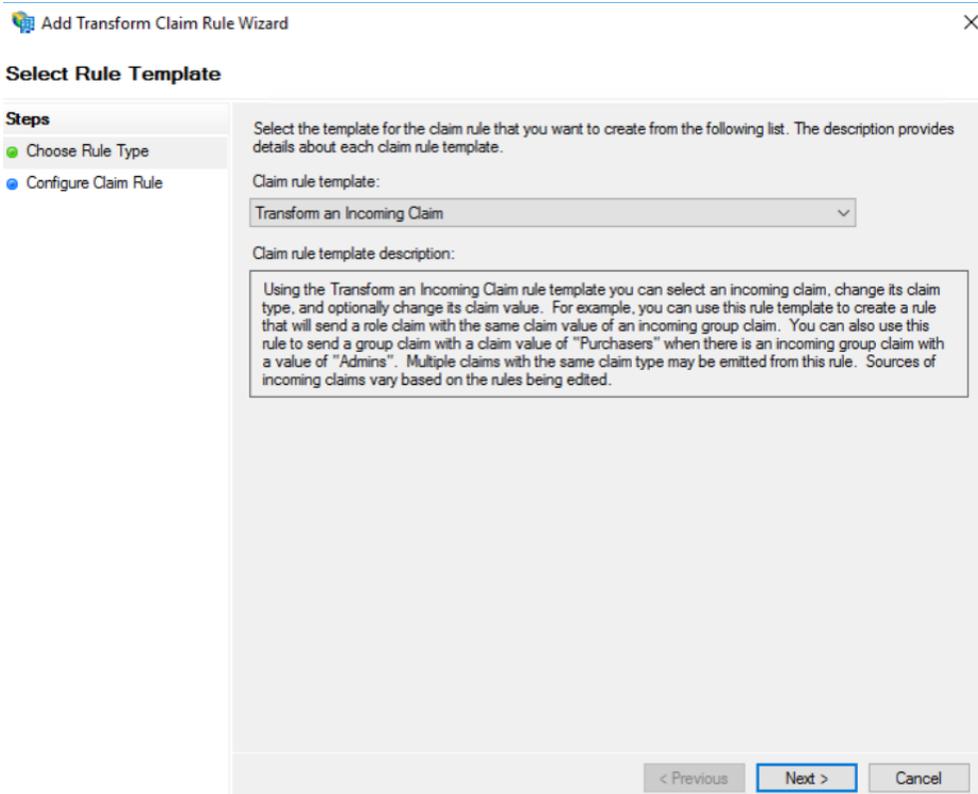
You need to define the claim issuance policy. Before you start you need to create the Relying Party Trusts as described in [Create Relying Party Trust in ADFS](#).

STEP 1 | From the right menu pane of the Relying Party Trusts, click **Edit Claim Issuance Policy**



STEP 2 | Click **Add Rule**.

STEP 3 | In the Add Transform Claim Rule Wizard, select **Transform an Incoming Claim** from the drop down list.



STEP 4 | Click **Next**.

STEP 5 | In the Configure Claim Rule page, type the Claim rule name **WindowsAccountName** which will pass the user login name in AD and select the Windows account name for the Incoming and Outgoing claim type.

STEP 6 | Click **Finish**.

STEP 7 | Add another claim rule which will pass the AD user account attributes to Cortex XSOAR. This step is required to map the user group membership, full name, email, phone and other LDAP attributes.

1. From the right menu pane of the Relying Party Trusts, click **Edit Claim Issuance Policy**
2. Click **Add Rule**.
3. In the Add Transform Claim Rule Wizard, select **Send LDAP Attributes as Claims** from the drop down list.
4. Click **Next**.
5. In the Configure Claim Rule page, type a claim rule name, select Active Directory from the Attribute store drop down list and map the required fields. Note that the user group attribute is mandatory if you wish to map the user group to the Cortex XSOAR user role.

Edit Rule - AD Attrbiutes ✕

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name:

Rule template: Send LDAP Attributes as Claims

Attribute store:

Mapping of LDAP attributes to outgoing claim types:

	LDAP Attribute (Select or type to add more)	Outgoing Claim Type (Select or type to add more)
▶	E-Mail-Addresses	E-Mail Address
	Given-Name	Given Name
	Surname	Surname
	Token-Groups - Unqualified Names	Group
	SAM-Account-Name	Windows account name

6. Click **Finish** and then click **OK** to create the claim rules.

STEP 8 | Open PowerShell and make sure the IDP Sign-on page is enabled

```
PS C:\Users\Administrator> Get-AdfsProperties | fl *init*

RelayStateForIdpInitiatedSignOnEnabled : True
EnableIdpInitiatedSignonPage           : True

PS C:\Users\Administrator> _
```

If one of these settings are set to false, enable it by typing `Set-AdfsProperties -<Property Name RelayState or EnableIdp> $True`

STEP 9 | Verify that the ADFS IDP Sign-on page is working by browsing to the ADFS service portal URL, in our example: <https://demistodev.local/adfs/ls/idpinitiatedsignon.aspx>

STEP 10 | Continue with [Configure the SAML 2.0 Integration for ADFS](#).

Configure the SAML 2.0 Integration for ADFS

You need to configure the SAML 2.0 integration so you can use it in Cortex XSOAR.

Before you start, ensure that you have defined the claim issuance policy as described in [Define the Claim Issuance Policy](#).

STEP 1 | Log in to the Cortex XSOAR server.

STEP 2 | Click **Settings > Integrations > Servers & Services**.

STEP 3 | Search for **SAML 2.0**.

STEP 4 | Click **Add Instance** to configure a new integration.

STEP 5 | Add the [SAML 2.0 ADFS Parameters](#).

STEP 6 | Select the following checkboxes:

- (Optional) Do not validate server certificate (insecure) - If you are using a self-signed certificate for the ADFS server you can select this checkbox.
- ADFS
- Compress encode URL (ADFS)

STEP 7 | Click **Test** to validate the URLs, token, and connection.

STEP 8 | To verify that the settings are successful, in the instance settings, click **Get service provider metadata**.

STEP 9 | [Map ADFS Groups to Cortex XSOAR Roles](#).

SAML 2.0 ADFS Parameters

The following table describes the SAML 2.0 parameters for ADFS, when adding a new instance in Cortex XSOAR:

Attribute	Description
Name	A name for the integration instance.
Service Provider Entity ID	The URL of your Cortex XSOAR server (also known as an ACS URL). In the format: <code>https://yourdomain.com/saml</code>
Idp metadata URL	ADFS URL+ <code>/federationmetadata/2007-06/federationmetadata.xml</code>
IdP SSO URL	ADFS URL+ <code>/adfs/ls/idpinitiatedsignon.aspx</code>
Attribute to get email	Attribute in your IdP for the user's email address.
Attribute to get user name	Attribute in your IdP for the user's user name.
Attribute to get first name	Attribute in your IdP for the user's first name.
Attribute to get last name	Attribute in your IdP for the user's last name.
Attribute to get groups	Attribute in your IdP for the groups of which the user is a member.
Default role	Role to assign to the user when they are not a member of any group. Users can be assigned to a default role at Cortex XSOAR in case there is no mapping between their AD group membership and a Cortex XSOAR server role.
Service Identifier (ADFS)	The ADFS relay identifier which Cortex XSOAR will redirect the user for SSO first login.

Map ADFS Groups to Cortex XSOAR Roles

Map the AD user account group memberships to a Cortex XSOAR server role.

STEP 1 | Go to **Settings > Users and Roles > Roles**.

STEP 2 | Create or edit an existing role, as described in [Define a Role](#).

STEP 3 | In the **SAML Roles Mapping** field, specify one or more SAML groups to map to the Cortex XSOAR role.

Configure User Notifications

Cortex XSOAR enables you to configure which notifications you want to receive and through which channels. Each user can define their own notification settings, so that a user's preferences does not affect another user.

Notifications are presented by categories. By default, all of the categories and channels are selected.

STEP 1 | Go to your user profile.

STEP 2 | Click the **Notifications** tab.

STEP 3 | Select the items for which you want to receive a notification, and the method through which you want to receive that notification, such as **Email** and **Mobile**.

STEP 4 | Click **Save**.

Set the Default Theme for New Users

By default, the color theme for new users is *light*. When you add the following server configurations, the selected theme is the one applied to all new user. Each user can change the theme in their user preferences.

STEP 1 | Go to **Settings > About > Troubleshooting**.

STEP 2 | In the **Server Configuration** section, click **Add Server Configuration**.

- Key: **default.theme**
- Value: Can be **dark**, **darkula**, or **light**

Disaster Recovery and Live Backup

- > Disaster Recovery and Live Backup Overview
- > Configure the Live Backup Environment
- > Transition a Standby Server to Active Mode
- > Transition an Active Server to Standby Mode
- > Transition Between DR States Through the Configuration File
- > Upgrade the Live Backup Environment
- > Cortex XSOAR Engines and Disaster Recovery
- > Backup the Database

Disaster Recovery and Live Backup Overview

Live Backup enables you to mirror your production server to a backup server. In a disaster recovery situation, you can easily convert your backup server to be the production server.

Server actions are mirrored in real-time. There might be pending actions due to high server load, connectivity issues, and so on. Note the following:

- Live Backup uses a single main server and a single standby server. Beyond these, additional servers are not currently supported.
- Active/Active configuration is not currently supported.
- Each host retains its own distinct IP address and host name.
- Neither host has any awareness of which node is truly active. Therefore, failover is not dynamic, meaning that making a node active must be done manually, by an administrator.

In the event of a server failover, engines dynamically reconnect to the active host.



As the process of making a Cortex XSOAR server active is a manual process, it is conceivable that two servers could be active simultaneously. You must avoid this scenario because both hosts collect and work on potentially the same security incidents, which could possibly lead to the following:

- *Incident duplication*
- *A higher load on your integration endpoints*
- *Possible significant database inconsistencies due to duplication of internal identifiers being shared between nodes and causing existing incidents to be overwritten.*



If there is ever uncertainty about whether a host that is presently down or stopped was in an active state before it went offline, it is recommended that you put the presently active node into a standby state before starting the Cortex XSOAR service on the other host. You can then make it active again after you have confirmed whether the host you are starting is already in active mode.

To configure the live backup environment, see [Configure the Live Backup Environment](#).

The following scenarios describe how to test, and deal with active server failures:

- [DR Scenario #1: Testing the DR Environment](#)
- [DR Scenario #2: Unrecoverable Active Server Failure](#)
- [DR Scenario #3: Unrecoverable Standby Server Failure](#)

When you first install the Cortex XSOAR server and starts for the first time, you can use a configuration file to transition between DR states, as described in [Transition Between DR States Through the Configuration File](#).

If you need to upgrade your live backup environment, see [Upgrade the Live Backup Environment](#).

For details about the relationship between engines and disaster recovery, see [Cortex XSOAR Engines and Disaster Recovery](#). For information about host names, DNS, and disaster recovery, see [Host Names, DNS, and Disaster Recovery](#).

Host Names, DNS, and Disaster Recovery

Consider the following about host names, DNS and DR:

- When configuring Live Backup, each Cortex XSOAR server should have its own unique host name and IP address
- You may require for analysts to always navigate to the same host name when accessing Cortex XSOAR. In this scenario, configure a separate DNS record which points to the active Cortex XSOAR server. In the event of a server failover, you are required to manually repoint this DNS record to the IP of the newly-active Cortex XSOAR server.
- It is critical that the TTL of the DNS record be set to a zero value. If it is higher, analysts are not able to access the active server using the shared host name until the TTL of the record expires and the DNS record is refreshed in the cache. This could take more than an hour.
- If you do not require a single URL to access Cortex XSOAR, when a server failover occurs, you might point your browser to the URL of the newly-active Cortex XSOAR server.

Configure the Live Backup Environment

Live Backup enables you to mirror your production server to a backup server, and in disaster recovery scenarios to easily convert your backup server to be the production server.



Before you start, ensure that you save the disaster recovery configurations before you copy all files

STEP 1 | Go to **Settings > About > Troubleshooting > Server Configurations** and do the following:

1. Verify that the **External Host Name** is correct.
2. Click **Add Server Configuration**.
3. Add the following key and value.

Key	Value
<code>ui.livebackup</code>	<code>True</code>

STEP 2 | Go to **Settings > Advanced > Backups** and in the **Live Backup** field, select **ON**.

STEP 3 | Configure the following backup server parameters.

Parameters	Value
Hostname/IP Address	Backup server IP address or Host name (without https:// prefix).
Port	Default is 443.
Trust server certificate (unsecured)	ON: certificates are not checked. OFF: certificates are checked.
Use proxy	Select whether to use a proxy.

STEP 4 | On another machine with a different host name or IP address, install Cortex XSOAR using the `-- -dr -do-not-start-server` flag, by typing the following command:

```
# ./demistosever-xxxx.sh -- -dr -do-not-start-server
```

STEP 5 | Verify that the backup server is accessible from the production server through port 443 (or any other port configured as a listening port). Ensure that there are no firewalls that might drop communication.

STEP 6 | Stop the Cortex XSOAR server, by typing the following command:

```
sudo service demisto stop
```

STEP 7 | Create a tarball file of the necessary files and folders on the production server to be copied to the backup server. Ensure that all files and folders have **demisto:demisto** ownership.

The following command preserves *demisto:demisto* ownership and file permissions:

```
# tar --ignore-failed-read -pczf demistoBackup.tgz /var/lib/demisto/data /  
var/lib/demisto/artifacts /var/lib/demisto/attachments /var/lib/demisto/  
systemTools /var/lib/demisto/d2_server.key /usr/local/demisto/cert* /usr/  
local/demisto/demisto.lic
```

STEP 8 | Copy the necessary files and folders from the production server to the backup server. Ensure all files and folders have **demisto:demisto** ownership.

- `data`
- `/var/lib/demisto/artifacts`
- `/var/lib/demisto/attachments`
- `/var/lib/demisto/systemTools`
- `/var/lib/demisto/d2_server.key`
- `/var/lib/demisto/cert*`
- `/var/lib/demisto/demisto.lic`

STEP 9 | Copy the created tarball file (`demistoBackup.tgz`) to the backup server using either `scp` or a tool that you prefer, by typing the following:

```
# scp demistoBackup.tgz root@<yourBackupServerIPorHostname>:/root
```

STEP 10 | On the backup server, extract the backup tarball file with the following command (original file permissions and ownership will be preserved).

```
# tar -C / -xzipvf demistoBackup.tgz
```

STEP 11 | Start the backup server and then the production server by typing the following command in each environment:

```
sudo service demisto start
```

If the procedure is successful, **Live Backup** is **ON**:

If the server is active, Cortex XSOAR appears as usual when you connect. You can [Transition an Active Server to Standby Mode](#) or [Transition a Standby Server to Active Mode](#).

Configure Live Backup for Multiple SAMLs

These steps describe how to configure live backup for multiple SAMLs by configuring SAML integrations for each server.

STEP 1 | Verify the **External Host Name** value matches the URL of the production server by going to **Settings > Troubleshooting > Server Configuration**.

STEP 2 | On the production server:

1. Go to **Settings > INTEGRATIONS > Servers & Services** and search for **SAML2.0**.
2. Create a SAML integration by clicking **Add Instance**.
3. Add the required parameters. For more information about setting up a SAML integration, see [Authenticate Users with SAML 2.0](#).

Ensure that **Service Provider Entity ID** parameter matches the URL of the production server. This is the only valid value. For example, `https://prod.demisto.com/saml`.

4. Verify that you can access the production server using SSO authentication.

STEP 3 | On the disaster recovery (DR) server:

1. Create a SAML integration by repeating steps 2.1 to 2.3.

Ensure that the **Service Provider Entity ID** parameter matches the URL of the DR server. This is the only valid value. For example, <https://dr.demisto.com/saml>.

2. Verify that you can access the production server.

STEP 4 | (Optional) For Dev/Test environments, switch to the DR server and the test SSO login.

DR Scenario #1: Testing the DR Environment

Before you make the backup server the new production server, ensure the original production server is down (not live). If you make the backup server the new production server while the original production server is still live, you can experience significant issues.

After finish testing the failover scenario (backup environment), you need to revert the server to its original state, which you can do through the UI or by modifying the configuration file.

STEP 1 | If you want to use Cortex XSOAR to backup the server, do the following:

1. On the live production server, select **Settings > Advanced > Backups > Switch Hosts**.
2. When prompted, complete the online **Switch Hosts** instructions.

Ensure that the production server is not live.

3. Go to the backup server and follow the on-screen instructions:



4. In the backup server environment, go to **Settings > About > Troubleshooting**.
5. For the External Host Name key, update the value to the host name of the backup server (the new production server).

The backup server is unaware of its external host name.

After a successful switch, the backup server is now live.

STEP 2 | If you want to use a configuration file to test the DR environment, do the following:

1. In the backup server, stop the server by typing:

```
sudo service demisto stop
```

2. Open the `/etc/demisto.conf` file and change the `Server.dr.enabled` property to `false`.
3. In the live server, open the `/etc/demisto.conf` file and change the `Server.dr.enabled` property to `true`.
4. Access the backup server using its IP address to check that the server is up and running.

STEP 3 | To revert to the original settings, repeat the steps above.

When following the instructions, remember the backup server is now the production server and production server is now the backup server.

DR Scenario #2: Unrecoverable Active Server Failure

In the event of an active server failure where the server cannot be restored (flooding, fire, meteor impact, hardware failure, etc), follow these steps to convert the backup server to the new production server and then configure a new backup server.

Before you start you need to [Configure the Live Backup Environment](#).

STEP 1 | On the standby server, follow the steps in [Transition a Standby Server to Active Mode](#).

STEP 2 | If your analysts use a single, pivoting host name to connect to the active Cortex XSOAR server, update your DNS record to re-point your Cortex XSOAR server host name to the now active server. For more information about host names, see [Host Names, DNS, and Disaster Recovery](#).

STEP 3 | If using engines, confirm that they are connected in **Settings > Integrations > Engines**.

If they have not reconnected and you have confirmed that network connectivity is good between the engine and the now active (previously backup) server (i.e., it is reachable on TCP 443 or the port you have configured), then follow the guidance in [Host Names, DNS, and Disaster Recovery](#).

STEP 4 | Obtain a new server environment according to your requirements.

Do not install Cortex XSOAR until step 5.

STEP 5 | Follow the procedure for [Configure the Live Backup Environment](#) using your now-active server as the primary host, and copying its files and data to the newly-built Cortex XSOAR server. Confirm that Live Backup is working.

STEP 6 | If appropriate for your environment (depends on whether you want to remain on the present active node), transition the active node over to the newly-built host by following the procedure [Transition an Active Server to Standby Mode](#) and confirming that Live Backup is again operational.

STEP 7 | If applicable, follow step 3 to reconfirm that engines are connected.

STEP 8 | Re-point your shared DNS record, if applicable, back to the primary Cortex XSOAR server and have analysts reconnect.

STEP 9 | Confirm that Cortex XSOAR is working by confirming that your integrations are working properly, incidents are being created normally, and that Analysts can login and work normally in Cortex XSOAR.

DR Scenario #3: Unrecoverable Standby Server Failure

In the event of an unrecoverable standby server failure (flooding, fire, meteor impact, hardware failure, etc), follow these steps.

Before you begin, ensure that you [Configure the Live Backup Environment](#).

STEP 1 | Obtain the new server that will serve as your new standby server according to your requirements.

Do not install until step 2.

STEP 2 | Follow the procedure for [Configure the Live Backup Environment](#) using the newly-active server as the primary host, and by copying its files and data to the newly-built Cortex XSOAR server. Confirm that Live Backup is working.

STEP 3 | Test the new server by following the steps in [DR Scenario #1: Testing the DR Environment](#).

Transition an Active Server to Standby Mode

If you are performing a manual failover in a DR simulation (when both hosts are operational), remember to always first put your active server into Standby mode before failing over.

STEP 1 | On the active server, navigate to **Settings > Advanced > Backups**.

STEP 2 | Click **Switch Hosts**.

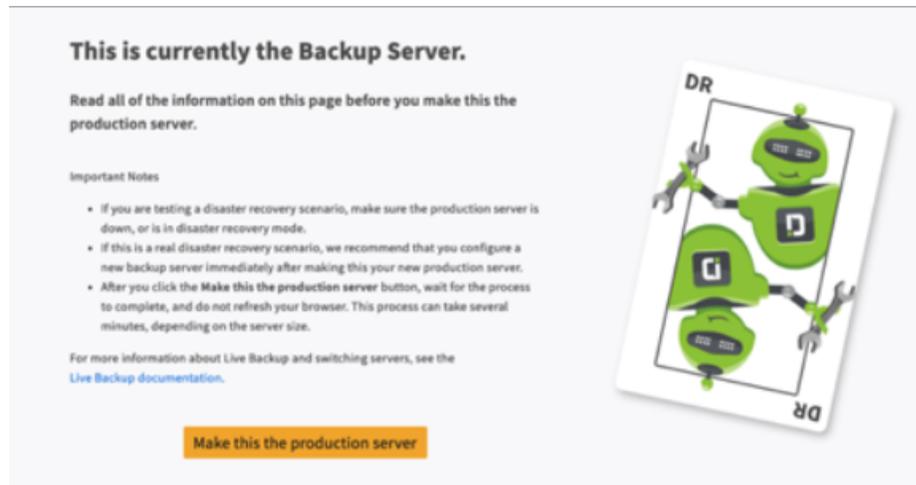
STEP 3 | In the Switch Hosts dialog box, in the text box, type **switch Hosts**.

STEP 4 | To commit the change, click **Yes, switch hosts**.

You are immediately taken back to the login page, where if you try to login again, you will be presented with the **This is currently the backup server page**.

Transition a Standby Server to Active Mode

If the server is active, when you connect to it, Cortex XSOAR appears as normal. After logging in, if the server is in standby mode, it appears like this:



STEP 1 | On the standby server being transitioned to active mode, from the This is currently the Backup Server page, click **Make this the production server**.

STEP 2 | In the Switch Hosts dialog box, in the text box, type `switch Hosts`.

Cortex XSOAR appears and the host is now active.

Transition Between DR States Through the Configuration File

It is possible to transition a server between active and standby states via the `/etc/demisto.conf` configuration file, when the server is new and starts for the first time. It cannot be used at any other time.

You need to set the `server.dr.enabled` configuration property to `true` or `false`. If set to `true`, the server is in DR mode when it starts. If set to `false`, the server is in active mode when it starts.

`/etc/demisto.conf`

```
{
  "Server": {
    "HttpsPort": "443",
    "dr": {
      "enabled": true
    }
  },
  "db": {
    "index": {
      "entry": {
        "disable": true
      }
    }
  }
}
```

If the DR state of the server is ever transitioned through Cortex XSOAR, the setting is stored in the Cortex XSOAR database rather than in the `demisto.conf` file. The database config setting for `dr.enabled` always takes precedence over `demisto.conf`, and changing the `server.dr.enabled` setting in the `demisto.conf` file has no effect.

In a real disaster recovery scenario, one in which the original production server is unrecoverable, you need to convert the backup server to the new production server and then configure a new backup server.

Upgrade the Live Backup Environment

Follow this procedure when upgrading your Cortex XSOAR version.

STEP 1 | Stop the main Cortex XSOAR server and the DR Cortex XSOAR server by typing the following command.

```
sudo service demisto stop
```

STEP 2 | Upgrade the DR Cortex XSOAR server.

STEP 3 | Upgrade the main Cortex XSOAR server.

STEP 4 | If the DR Cortex XSOAR server did not start, restart the DR Cortex XSOAR server.

STEP 5 | If the main Cortex XSOAR server did not start, restart the main Cortex XSOAR server.

Cortex XSOAR Engines and Disaster Recovery

In the event of a failover between the Cortex XSOAR server, engines are capable of dynamically failing over to the active node. This should happen automatically if the engine was deployed after DR was configured.



Assuming all is configured and working properly, it should not be necessary to change DNS to affect an engine failover when a server failover occurs.



*If engine failover is not working when failing over between Cortex XSOAR servers (i.e., does not display as **Connected: false** in **Settings > Integrations > Engines**, it is likely due to one of the following causes:*

- *The file `/var/lib/demisto/d2_server.key` is not the same on each Cortex XSOAR server. This can sometimes happen if Live Backup was previously configured using Cortex SOAR (Demisto) 4.0 and this file did not exist at the time that Cortex XSOAR was first configured. Copy this file from the primary server to the backup server and restart the backup server service.*
- *On the engine, the **EngineURLs** array property of `/usr/local/demisto/d1.conf` is missing the IP or host name of the backup Cortex XSOAR server. Solutions are:*

*Simply redeploy the engine from **Settings > Integrations > Engines**. This should automatically include both servers in the `d1.conf` file.*

*Modify the JSON in conf file manually, to add the other server to the **EngineURLs** array, and restart the engine. If a syntax error is detected in the JSON, the engine service refuses to start and may not log any error messages. The array should now look something like:*

```
EngineURLs": [
  "wss://cortex_xsoarserver1:443/d1ws",
  "wss://cortex_xsoarserver2:443/d1ws"
],
```

- *Host name resolution is broken from the engine to one of your servers. Use **ping** or **nslookup** to confirm that the engine host can resolve the backup server, and that the IP address is of the server correct. If not, it may require a change to your DNS environment or a network or host firewall is blocking connectivity from the engine to your backup Cortex XSOAR server.*

Backup the Database

Cortex XSOAR backs up the database on a daily basis, by storing the entire database of incidents, playbooks, scripts, and user defined configurations.

You can define whether you want Cortex XSOAR to create automatic backups, and the location to store the backups. The database backup files are located in `/var/lib/demisto/backup`.

Every day Cortex XSOAR adds to its store of daily, weekly, and monthly backups files. You can define the maximum number of recent backups to store.

STEP 1 | Back up the database.

1. Select **Settings > ADVANCED > Backups > Automated Backups**.
2. In the **Backup Directory**, type the location for the backups directory.
3. In the **Backup Time** directory, schedule backup time of day.
4. Define the maximum number of daily, weekly, and monthly backups to keep.

It is recommended that you make a backup copy of the backup directory standard backup tools and save them on a different machine.

STEP 2 | Restore the database.

1. Log out all users from Cortex XSOAR.
2. Stop the service.
3. Delete the content of the database directory.

By default, the database directory is `/var/lib/demisto/data`.

4. Copy the backup file to the database location.
5. Extract the `.gzip` backup file using `tar -xzf <file-name>`.
6. Move the `demisto_XXXXX.db` files to the `partitionsData` folder. Keep the `demisto.db` file in the `/data` parent folder.
7. Restart the server and log in to Cortex XSOAR.

STEP 3 | Backup additional directories.

In addition of the database, the following directories need to be backed up and restored manually.

- `/var/lib/demisto/artifacts`
- `/var/lib/demisto/attachments`
- `/var/lib/demisto/d2_server.key`
- `/var/lib/demisto/tools`
- `/var/lib/demisto/versionControlRepo`
- `/usr/local/demisto`
- `/etc/demisto.conf`

To restore the files return them to the server with the flag `-restore-indexing=true`.

The re-index process might take a while.

Remote Repositories in Cortex XSOAR

- > Remote Repositories Overview
- > Configure a Remote Repository on a Development Machine
- > Configure a Remote Repository on the Production Machine
- > Edit and Push Content to a Remote Repository
- > Troubleshoot a Remote Repository Configuration

Remote Repositories Overview

Cortex XSOAR supports the ability to work with separate repositories for development and production environments. This enables you to develop and test all of your content in one location, and when it is ready, you push the content to the remote repository. On your production environment, you pull the content as you would all other content updates.

In addition, Cortex XSOAR content updates are only delivered to the development environment. This enables you to determine which updates you want to push to production.



Working with remote repositories is git-based. Any service that supports this protocol can be used, for example, GitHub, GitLab, Bitbucket, etc. In addition, on-premise repositories are also supported.

How it Works

In the production environment, the content appears as a content update, just like any other, and you pull the content from the remote repository into your working branch.

To work with remote repositories, you must have two separate Cortex XSOAR environments on two separate machines. The development environment is used to write the following content:

- Automations
- Playbooks
- Integrations
- Classification
- Agent tools
- Incident fields
- Indicator fields
- Evidence fields
- Incident layouts
- Incident types
- Pre-processing rules



If you have more than two pre-processing rules in your Local Changes queue, you must push all of those changes to the remote repository.

- Indicator types
- Reports
- Dashboards
- Widgets

On the production environment, it is not possible to edit these elements.

You need to configure a remote repository both on a [development environment](#) and a [production environment](#). After you develop your content, if you want it to be available as part of a content update for the production environment, you must [push](#) the changes to the remote repository. If you experience issues, learn how to [troubleshoot remote repositories](#).

Configure a Remote Repository on a Development Machine

To work with the remote repositories, you must have at least two machines, one for development and one for your [production machine](#).

Prerequisites

Before configuring the remote repository, review the following list of prerequisites:

- Ensure that you have network connectivity from the Cortex XSOAR server to your repository. All communication goes through the Cortex XSOAR server so it must have access to the remote repository.



You cannot configure a Cortex XSOAR engine to manage communication to the remote repository.

- When creating a repository in your remote Git platform, verify that the repository contains branches. Defining the repository in Cortex XSOAR does not create the branches.
- Before toggling the remote repository feature on or off, or changing your repository configuration, ensure to back up your existing content to your local computer by navigating to **Settings > About > Troubleshooting > Custom Content** and clicking **Export**.
- You need to install Cortex XSOAR on your development environment, as described in [Single Server Deployment](#).

STEP 1 | Configure the environment.

1. On the machine that is designated to be the development environment, navigate to **Settings > About > Troubleshooting > Custom Content** and click **Import**.
2. Verify that the content in both the development and production environments is synchronized before you proceed. If you do not synchronize the content, you might lose content in one of the environments.

STEP 2 | Define the Repository.

1. Go to **Settings > Advanced > Content Repository**.
2. Click the On/Off slider to enable the remote repositories
3. Set the current machine as the development environment.
4. Enter the URL and connection credentials to the repository. Only SSH connections are supported.

If your SSH connection uses a port other than port 22 (the default SSH port), you must include the ssh string and port number in the url. In the following example, we use port 20017:

```
git@content.demisto.com:20017/~ /my-project.git
```

5. Select the active branch on which you will be working.
6. Click **Save**.
7. In the Migrate server changes screen, determine whether or not you want to keep the content that is currently on the development server, or discard the changes and synchronize completely with the remote repository.

Selected content: the current content on your server will be maintained and presented in the Local Changes window.

Cleared content: the current content on the server will be overwritten by the content in the remote repository, if it exists.

Migrate server changes



The following changes will be migrated to the new configuration.

By clicking Continue, the selected changes will persist in the new configuration. Unselected changes will be discarded, and as a result, some functionality might not be available.

<input checked="" type="checkbox"/>	Name ↑	Type	Status
<input checked="" type="checkbox"/>	AbuseIPDB	Integration	Modified
<input checked="" type="checkbox"/>	AbuseIPDBPopulateIndicators	Automation	Modified
<input checked="" type="checkbox"/>	Access Details	Layout	Modified
<input checked="" type="checkbox"/>	Access Edit	Layout	Modified
<input checked="" type="checkbox"/>	Access Investigation - Generic	Playbook	Modified
<input checked="" type="checkbox"/>	Access Investigation - QRadar	Playbook	Modified

Cancel

Continue

8. Click **Continue**.

Content from the remote repository is installed.

This can take several minutes depending on the amount of content in the remote repository and your hardware configuration. Your custom content is automatically backed up to the Cortex XSOAR server any time you change one of the remote repository settings. The backup is located under `/var/lib/demisto/backups/content-backup-*.tar.gz..`

STEP 3 | Manage Classifiers.

1. Go to **Settings > Integrations > Classification and Mapping**.
2. Select the classifier that you want to keep.
3. Make a small change and save the classifier.

Each integration instance has its own classification. You can only push one classification per integration. When switching the remote repository to dev mode, if you have multiple instances of the same integration, the last classification that was saved appears in the Local Changes. You should decide which classification you want to keep, and make a small change to that classification

STEP 4 | Push Content to the Remote Repository.

1. Go to **Settings > Local Changes**.
2. Select the changes that you want to push to the remote repository, and click **Push**.

INTEGRATIONS USERS AND ROLES ADVANCED **LOCAL CHANGES** ABOUT

Showing 564 local changes by [Everyone](#) of type [Automation](#)

Search in table...

Push

<input type="checkbox"/>	Name	Type	Change	By	Changed ↓
<input type="checkbox"/>	SendMessageToUser	Automation	Modified	Admin Dude	March 17, 2019, 5:38 PM
<input type="checkbox"/>	ConferIncidentDetails	Automation	New	DBot	March 17, 2019, 5:28 PM
<input type="checkbox"/>	CybereasonPreProcessingExample	Automation	New	DBot	March 17, 2019, 5:28 PM
<input type="checkbox"/>	IncidentSet	Automation	New	DBot	March 17, 2019, 5:28 PM
<input type="checkbox"/>	MarkAsNoteByTag	Automation	New	DBot	March 17, 2019, 5:28 PM

You should not manually push content to the remote repository. Use only the procedures outlined in the documentation to ensure that your content is properly updated in the production environment.

Configure a Remote Repository on the Production Machine

To work with the remote repositories, you must have at least two machines, one for your [development environment](#) and the other for your production environment.

Prerequisites

Before configuring the remote repository, review the following list of prerequisites:

- Ensure that the Selective Propagation feature is enabled (default).
- Verify that the content that you want already exists in the remote repository. You cannot edit content in the production environment. All of the content is fully synchronized with the remote repository.



Any content that exists in the production environment, but not on the remote repository, will be deleted

STEP 1 | Define the repository.

1. Navigate to **Settings > Advanced > Content Repository**.
2. Click the On/Off slider to enable the remote repositories.
3. Define the machine as the production environment.
4. Enter the URL and connection credentials to the repository. Only SSH connections are supported.

If your SSH connection uses a port other than port 22 (the default SSH port), you must include the ssh string and port number in the url. In the following example, we use port 20017:

```
git@content.demisto.com:20017/~ /my-project.git
```

5. Select the active branch from which you pull content.
6. Click **Save**.

In the Discard server changes screen, you are presented with content that exists in your production environment, but does not exist on the remote repository. This includes integrations, and their instances and classifiers.

Discard server changes
✕

Content

You have content in your current configuration that does not exist in the new configuration (e.g. integrations, playbooks, etc.)
By clicking continue, you will lose all of the changes you made to this content and some functionality might not be available as a result.

Name	Type	Status
ADExpirePassword	Automation	Deleted
ADGetAllUsersEmail	Automation	Deleted
ADGetCommonGroups	Automation	Deleted
ADGetComputer	Automation	Deleted
ADGetComputerGroups	Automation	Deleted
ADGetEmailForAllUsers	Automation	Deleted

Integrations

The following 4 integrations and their instances will be deleted

- Image OCR (1 instance)
- Palo Alto Networks Cortex (2 instances)
- Rasterize (1 instance)
- Where is the egg? (1 instance)

Type "Discard" to proceed *

Cancel
Continue

7. Type **Discard** in the relevant field and click **Continue**. All of the content that appears in this screen is discarded and permanently deleted.

Content from the remote repository is installed. This can take several minutes depending on the amount of content in the remote repository and your hardware configuration. In addition, your custom content is automatically backed up to the Cortex XSOAR server any time you change one of the remote repository settings. The backup is located under `/var/lib/demisto/backups/content-backup-*.tar.gz`.

STEP 2 | Pull Content from the Remote Repository

Once you push content from the Development machine to the remote repository, it is available as an update for the production environment.

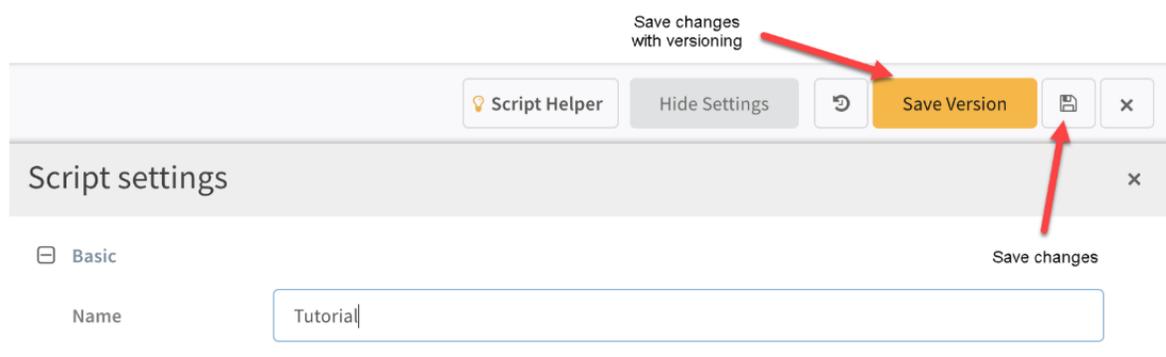
1. Check for updates.
2. Click **Install Content**.

Edit and Push Content to a Remote Repository

Once you develop your content, for it to be available as part of a content update for the production environment, you must push the changes to the remote repository.

 *You should not manually push content to the remote repository. Use only the procedures outlined in the documentation to ensure that your content is properly updated in the production environment*

You can save version and manage revisions locally using the Save Version button. Alternatively, you can click and save the changes.



These options are only available for the following content types:

- Automations
- Playbooks
- Integrations
- Classifications
- Layouts
- Reports
- Dashboards

For all other content types, your changes are automatically saved to the local changes.

Best Practices

To avoid overwriting content on your local development server, Cortex XSOAR recommends that you work with a separate branch for each development environment.

If you do not work with separate branches, when trying to push content to an out-dated remote branch, you will receive a pull changes message.



Accepting this message overwrites the content on your local development server.

STEP 1 | Push Content to Remote Repository.

1. Navigate to **Settings > Local Changes**.

The available content changes are shown.

2. Select the changes that you want to push to the remote repository, and click **Push**.

When using remote repositories in a multi-tenant environment, the main account uses propagation labels to distribute content to accounts. These labels must come with the content that is being pushed from the development environment. When editing your content in the development environment, keep in mind that you must also add propagation labels as part of the editing.

<input type="checkbox"/>	Name	Type	Change	By	Changed ↓
<input type="checkbox"/>	SendMessageToUser	Automation	Modified	Admin Dude	March 17, 2019, 5:38 PM
<input type="checkbox"/>	ConferIncidentDetails	Automation	New	DBot	March 17, 2019, 5:28 PM
<input type="checkbox"/>	CybereasonPreProcessingExample	Automation	New	DBot	March 17, 2019, 5:28 PM
<input type="checkbox"/>	IncidentSet	Automation	New	DBot	March 17, 2019, 5:28 PM
<input type="checkbox"/>	MarkAsNoteByTag	Automation	New	DBot	March 17, 2019, 5:28 PM

STEP 2 | Control Access for Pushing Content.

1. Navigate to **Settings > About > Troubleshooting > Server Configuration**
2. Click **Add Configuration**.
3. Under key, enter **UI.version.control.admin.only**
4. Under value, enter **true**.
5. Click **Save**.

Troubleshoot a Remote Repository Configuration

You can troubleshoot the following issues:

- [Troubleshoot a Remote Repository Definition](#)
- [Troubleshoot Editing and Pushing Content](#)
- [Troubleshoot Content Issues](#)

FAQ

Question: Which services are supported?

Answer: Working with remote repositories is git-based. Any service that supports this protocol can be used, for example, GitHub, GitLab, Bitbucket, etc. In addition, on-premise repositories are also supported.

Question: Does Cortex XSOAR remote repository configuration support GPG signing?

Answer: yes

1. Connect to the remote repository using SSH.
2. Run the following commands:
 - `git--global config user.signingKey KEYID` where KEYID is your key ID.
 - `git --global config commit.gpgSign true`

Question: Can I limit permissions for pushing content?

Answer: Yes. See [Edit and Push Content to a Remote Repository](#).

Question: Can I edit content in my production environment?

Answer: No. When working in a remote repository configuration, content can only be edited in the development environment.

Troubleshoot a Remote Repository Definition

The following instructions describe how to troubleshoot when defining a remote repository.

dial TCP Error Message

If a `dial tcp...` error message appears, when defining your remote repository and loading the list of repository branches.

1. Contact your system administrator to enable connectivity. This may occur due to connectivity issues, such as a closed port or a proxy that is not enabling the connection.

Invalid SSH URL Error

An Invalid SSH URL error label appears, when defining your remote repository and entering the Repository URL.

1. Ensure the URL is not the HTTPS version. Currently only SSH connections are supported (and only from the server itself, no engine support).
2. Ensure the URL ends with `.git`` such as `ssh://git@content.demisto.com:20017/~my-project.git``.

Permission Denied Error

A permission denied error appears, when attempting to fetch the repository branches. This occurs because currently Cortex XSOAR supports only `RSA PRIVATE KEY``.

1. Ensure you have set the correct passphrase and are using the correct key type.
2. Navigate to the directory in which you saved the private key file.
3. Open the private key file in a text editor and verify that the file begins with `-----BEGIN RSA PRIVATE KEY-----`.

If the file does not begin with this text, regenerate the private key and add the `-m pem` flag. For example, `ssh-keygen -t rsa-b 4096 -C "your_email@example.com" -m pem`.

Internal Server Error

The `internal server error. Something went wrong.` error message appears, when attempting to fetch the repository branches.

1. In the server log files, search for `Host keyverification failed` error message.
2. Connect to the server by SSH.
3. Run the following command:

```
ssh-agent bash-c "ssh-add ~/key; git clone [url]"
```

4. Copy the file `~/.ssh/known_hosts` to `/var/lib/demisto/`.

Troubleshoot Editing and Pushing Content

When you push your content to the remote repository, you might receive the following error messages

Invalid Signatures When Committing

When you push your content to the remote repository, you might receive the following error message: `Error: Commits must have valid signatures.`

This occurs when your Git server configuration requires that every commit must have a signature.

1. Connect to the remote repository server using SSH.
2. Run the following commands:
 - `git --global config user.signingKey KEYID` where KEYID is your key ID.
 - `git --global config commit.gpgSign true`

GIT Version Out-of-Date

When you push your content to the remote repository, you might receive the following error message: `error GIT: failed executing [rebase [--exec=git log --max-count=1 --pretty=format:%H;printf '>';grep -o "\w\{40\}" .git/rebase-merge/done | tail -n 1 --strategyrecursive --strategy-option theirs --onto stag --root stag_workspace]]`

This occurs when you have an outdated GIT version on your server. You must have at least version 2.21.0.

To update your GIT version:

1. Export the custom content by going to **Settings > About > Troubleshooting > Custom content > Export**.
This exports the state of the current content.
2. Turn off the remote repository feature on the dev server.
3. On your server, either download and install an updated RPM package of the GIT client, or run the Demisto installer again.
4. Turn on the remote repository feature on the dev server.
5. Import the previously exported content.

This restores the local content.

Troubleshoot Content Issues

After enabling the remote repository feature or updating content, some content is missing. To resolve this issue, restore the content backup package that was created when you enabled the feature or updated content.

1. Copy the content package from the server to your local machine using the `scp` command.

The backup is located at `/var/lib/demisto/backup/content-backup-*`

2. In Cortex XSOAR, navigate to **Settings > About > Troubleshooting > Custom content > Import** and upload the content package.

Engines

- > [Cortex XSOAR Engines Overview](#)
- > [Install Cortex XSOAR Engines](#)
- > [Use an Engine in an Integration](#)
- > [Manage Engines](#)
- > [Configure Engines](#)
- > [Remove an Engine](#)
- > [Troubleshoot Cortex XSOAR Engines](#)

Cortex XSOAR Engines Overview

Cortex XSOAR engines are installed in a remote network and allow communication between the remote network and the Cortex XSOAR server. Although you cannot run scripts, you can run integration commands. It is possible to install a single engine, or multiple engines. An engine is used for the following purposes:

- [Proxy](#)
- [Load-Balancing](#)

Engine Proxy

Cortex XSOAR engines enable the Cortex XSOAR server to access internal or external services that are otherwise blocked by a firewall or a proxy, etc. For example, if a firewall blocks external communication and you want to run the Rasterize integration, you need to install an engine to access the Internet.

Engine Architecture

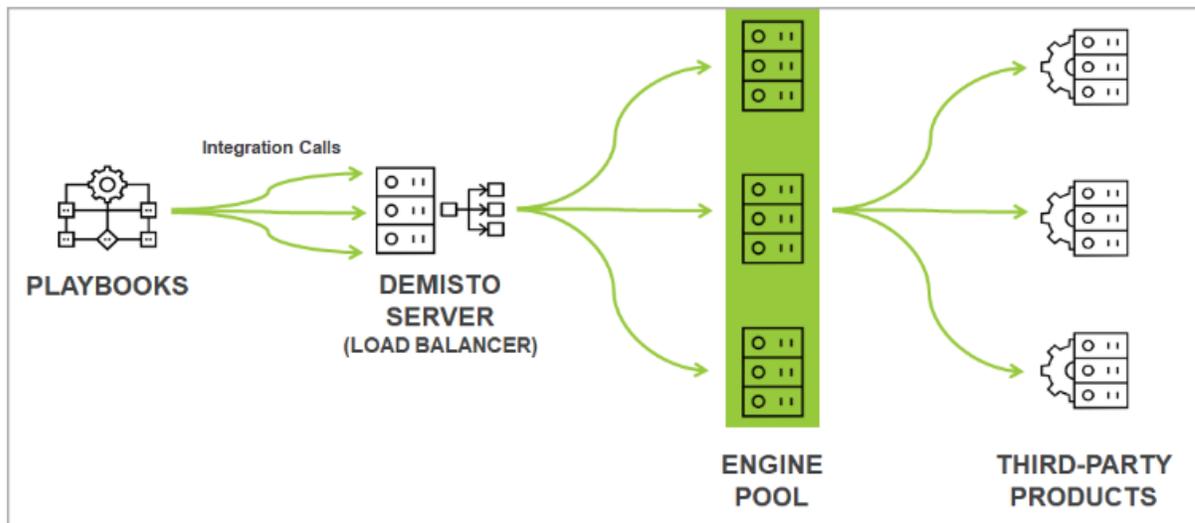


Within the network, you need to allow the Engine to access the Cortex XSOAR server's IP address and listening port (by default, TCP 443).

Engine Load-Balancing

Engines can be part of a load-balancing group, which enables distribution of the command execution load. The load-balancing group is a group of engines that use an algorithm to efficiently share the workload for integrations that the group is assigned to, thereby speeding up execution time. In general, heavy workloads are caused by playbooks that run a high number of commands.

Before configuring an integration to run using multiple engines in a load-balancing group, we recommend that you test the integration using a single engine in the load-balancing group.



 *When you add an engine to a load balancing group, you cannot use that engine separately. The engine does not appear in the engines drop-down menu when configuring an integration instance.*

By default, the Cortex XSOAR server is part of the load balancing group. It is recommended that you [Remove the Cortex XSOAR Server From the Load-Balancing Group](#) when there are two or more engines in the load-balancing group, or if you use engines to access integrations that are inaccessible from the Cortex XSOAR server.

Engine Installation and Configuration

You can [Install Cortex XSOAR Engines](#) on Linux and Windows machines. After installing the engine, you can [Configure Engines](#), such as log levels, remove a server to the load-balancing group, use as a web proxy, etc. You can also [Manage Engines](#), such as getting logs, add/remove engines, delete engines, remove engines, etc.

 *For Linux machines, you need to install [Docker](#) before installing an engine. If you use the Shell installer, Docker is automatically installed.*

Install Cortex XSOAR Engines

You can install Cortex XSOAR engines on all Linux and Windows machines. Although Cortex XSOAR engines are intended for Linux operating systems they can be used on Windows, but Windows machines do not support Python integrations

After creating an engine in the **Settings** page, you download one of the following file types for installation on the engine machine:

- **Shell:** For all Linux deployments, including Ubuntu, CentOS and RHEL, etc. Automatically installs Docker, downloads Docker images, and enables remote engine upgrade.

The installation file is selected for you. Shell installation supports the purge flag, which by default is false.

- **DEB:** For Ubuntu operating systems.
- **RPM:** CentOS and RHEL operating systems.

 Use DEB and RPM installation when shell installation is not available. You need to install Docker and any dependencies.

- **Zip:** Used for Windows machines.
- **Configuration:** Configuration file for download. When you install one of the other options, this config file (d1.conf) is installed on the engine machine.

 When you create the engine, the `python.engine.docker` key is set to `true`. If Docker is not available when the engine is created, the key is set to `false`. If this happens, in the `d1.conf` file, you need to set the key to `true`

Before you begin, check you have the required [Cortex XSOAR Engine](#) requirements.

After you install and deploy an engine, there are several ways that you can [Manage Engines](#). For Linux systems, you can run Python integrations on an engine. Ensure you have Python 2.7 or later installed on the engine machine. Running Python integrations needs to be through Docker.

STEP 1 | Define the base URL.

The base URL is the external IP address of your Cortex XSOAR server. If you do not define the base URL, you need to add it to the `d1.conf` file after you create the engine.

1. Go to **Settings > About > Troubleshooting**.
2. From the **Server Configuration** section, in the **Base URL (for D2 Agents and Engines)** type the Base URL.

For example, for `https://ec2-54-228-48-128.eu-west-1.compute.amazonaws.com/`, type `eu-west-1.compute.amazonaws.com`

STEP 2 | Create an engine.

1. Select **Settings > Integrations > Engines > Create New Engine**.
2. In the **Engine Name** field, add a meaningful name for the engine.
3. Select one of the installer types from the drop down list.

For Linux systems it is recommended to use the Shell installer.

4. **(Optional)** If you want to add the engine to a load balancing group, from the drop down list, select the group you want to add.

The dropdown list only appears after you have created and connected an engine and created a load balancing group.

The engine cannot be used as an individual engine and does not appear when configuring an engine from the drop down. list. To add the engine to a new load balancing group, see [Manage Engines](#).

5. (Optional) Select the checkbox to enable multiple engines to run on the same machine.
6. (Optional) Add any required configuration in JSON format.
7. Click **Create New Engine**.

STEP 3 | For Shell installation, do the following:

1. Move the `.sh` file to the engine machine using a tool like SSH or PuTTY.
2. On the engine machine, grant execution permission by running the following command:

```
chmod +x /<engine-file-path>
```

3. Run as a root user
4. Install the engine by typing one of the following commands:

With tools: `sudo <engine-file-path>`

Without tools: `sudo <engine-file-path> -- -tools+false`

If you receive a `permissions denied` error, it is likely that you do not have permission to access the `/tmp` directory. Run the `./demisto.sh --target {/targetFolder}` command. The `{targetFolder}` can be any folder to which you have write access.

STEP 4 | For RPM/DEB installation do the following:

1. Move the file to the required machine using a tool like SSH or PuTTY.
2. Type one of the following installation commands:

Machine Type	Install Command
CentOS/RHEL (RPM)	<code>sudo rpm -Uvh d1-2.5_15418-1.x86_64.rpm</code>
Ubuntu (DEB)	<code>sudo dpkg --install d1_xxx_amd64.deb</code>

3. Start the engine by running one of the following commands:

Machine Type	Start Command
CentOS/RHEL (RPM)	<code>sudo systemctl start d1</code>
Ubuntu (DEB)	<code>sudo service d1 restart</code>

STEP 5 | For zip file installation, do the following.

1. Move the `d1 zip` file to the engine machine using a tool like [WinSCP](#).
2. Unzip the file and move it to any location you require.
3. Open the file and run the `d1_windows_amd64.exe` file.

Every time you want to connect to Cortex XSOAR you need to run the D1 Application file. Alternatively, you can [run the engine as a service](#).

STEP 6 | Use an Engine in an Integration.

STEP 7 | (Optional) If you experience performance issues you may need to [Configure the Number of Workers for the Server and Engine](#).

Run the Engine as a Service on Windows

You can run the engine as service on Windows machines. In this procedure we use [NSSM](#), which is a free utility that manages background services and processes.

You can configure the [Run Registry Keys](#) instead of using the NSSM utility.

Before you begin, [install the engine](#) on the required machine.

STEP 1 | Download the latest NSSM zip file by going to <http://nssm.cc/download>.

STEP 2 | Unzip the file and move the contents to the same directory you installed the engine.

STEP 3 | Open a command prompt as an administrator and type the following command:

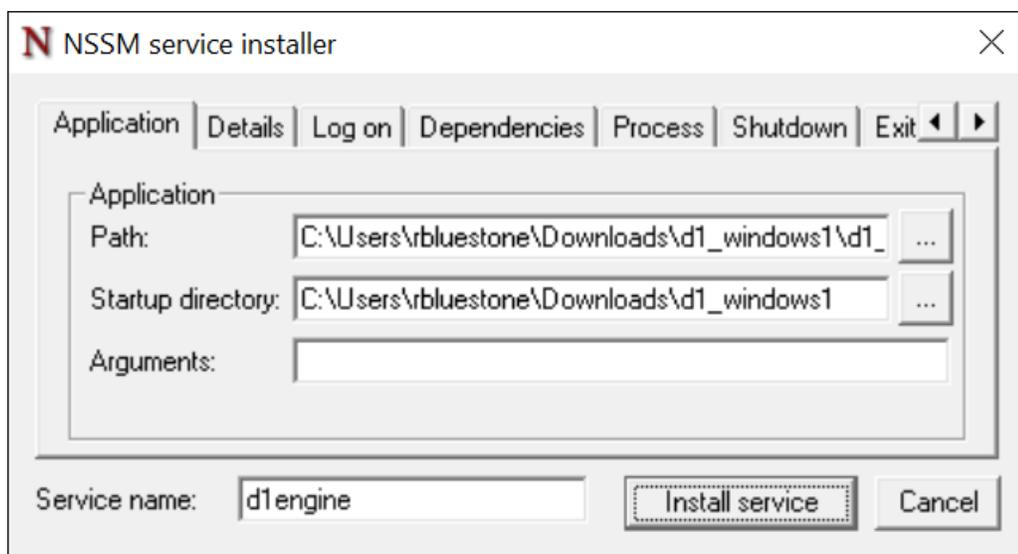
```
.\nssm.exe install dlengine
```

Ensure that the `nssm` exe file is in the same directory as the engine you want to run.

The NSSM service installer appears.

STEP 4 | In the **Application Path**, browse to the engine file.

The engine file is called `dl_windows_amd64.exe`. The **Startup directory** is automatically populated (where the `.conf` and `.log` files are located).



STEP 5 | Click **Install service**.

A message appears confirming that service was installed.

STEP 6 | Start the service, by typing the following command:

```
.\nssm.exe start dlengine
```

Confirmation appears that the engine started successfully.

STEP 7 | (Optional) Go to the **Task Manager** and check that the service is running.



STEP 8 | (Optional) To stop the service, type the following command:

```
.\nssm.exe stop d1engine
```

STEP 9 | (Optional) To remove the service, type the following command:

```
.\nssm.exe remove d1engine confirm
```

Use an Engine in an Integration

When you create an integration instance, you can select whether to fetch incidents and run commands executed for the integration using the engine or a load-balancing group of engines. After you assign the Engine or load-balancing group to an integration instance, you can run commands using the Engine or load-balancing group by specifying the using argument.

Command Example

```
!url url="www.cnn.com" using=urlscan.io_instance_1
```



Manage Engines

After you have installed the engine, you can manage engines and load-balancing groups by going to **Settings > Integrations > Engines** to manage engines and load-balancing groups. You can do the following:

- Add/remove engines to the [load-balancing group](#).

You can only add the engine to the load-balancing group after you have connected the engine

- Get engine logs

Logs are located in `/var/lib/demisto`.

- Upgrade an engine

When an engine requires an upgrade, the Cortex XSOAR version for that engine is red. You need to select the checkbox for the engine that requires an upgrade and click **Upgrade Engine**. When the upgrade finishes, the version appears in the **Cortex XSOAR Version**. The upgrade procedure can take several minutes.

You can only upgrade the engine if you installed the engine with the shell installer. To upgrade engines that were not installed with the shell installer, you need to remove the engine and do a fresh install. For more information, see [Install Cortex XSOAR Engines](#). For troubleshooting, see [Troubleshoot Engine Upgrades](#).

- Delete engines

Configure Engines

You can [Edit the Engine Configuration](#) either by modifying the `d1.conf` file on the engine or in Cortex XSOAR when [managing engines](#). You can only configure an engine in Cortex XSOAR if you have installed the engine using the shell installer.

You can configure the server and engine to do the following:

- [Configure the Number of Workers for the Server and Engine](#)
- [Configure Access to Communication Tasks through an Engine](#)
- [Notify Users When an Engine Disconnects](#)
- [Remove the Cortex XSOAR Server From the Load-Balancing Group](#)

Edit the Engine Configuration

You can edit the engine configuration by modifying the `d1.conf` file on the engine, or in the Cortex XSOAR platform by modifying specific properties in the **JSON formatted configuration** dialog box (Shell installations only).

STEP 1 | To modify the `d1.conf` file:

1. On the machine on which you installed the engine, navigate to the `d1.conf` file:

Installation Type	Location
RPM, DEB, Shell	<code>/usr/local/demisto</code>
ZIP	Same folder as the binary.

2. Modify the file as required. See [Common Properties When Editing an Engine Configuration](#).

You can [Configure the Engine to Use a Web Proxy](#) or [Configure the Engine to Call the Server Without Using a Proxy](#).

STEP 2 | Modify the configuration in Cortex XSOAR.

Ensure that the data is in JSON format. The properties that you specify override the values defined in the `d1.conf` file. A use case for modifying the engine configuration is if you want to generate engine logs for a specific log level.

1. From the engines table, select the engine you want to modify the configuration, and click **Edit Configuration**.
2. In the **JSON formatted configuration** section, modify the properties as required. For more information, see [Common Properties When Editing an Engine Configuration](#)

JSON formatted configuration

```
{
  "LogLevel": "debug",
  "LogFile": "d1new.log"
}
```

Common Properties When Editing an Engine Configuration

The following table describes the common properties when editing an engine configuration using the `d1.conf` file (located by default at `/usr/local/demisto/` in the **JSON formatted configuration** section).

Property	Type	Values	Edit
<code>http_proxy</code>	String	The IP address of the HTTP proxy through which the engine communicates.	The engine <code>d1.conf</code> file
<code>https_proxy</code>	String	The IP address of the HTTP/s proxy through which the engine communicates.	The engine <code>d1.conf</code> file
<code>LogLevel</code>	String	<ul style="list-style-type: none"><code>debug</code><code>info</code><code>warning</code>	The engine <code>d1.conf</code> file or in the JSON formatted configuration dialog box.
<code>BindAddress</code>	String	The port on which the engine listens for agent connection requests and communication task responses	The engine <code>d1.conf</code> file.
<code>EngineURLs</code>	String array	An array of Server addresses to which the engine tries to connect. If you change the server URL, you need to update this parameter.	The engine <code>d1.conf</code> file.
<code>LogFile</code>	String	Path to the <code>d1.log</code> file. If you change the name or location of the <code>d1.log</code> file, you need to update this parameter.	The engine <code>d1.conf</code> file
<code>engine.allow.data.connection</code>	String	Disables the option to send communication task forms through the engine. <ul style="list-style-type: none"><code>false</code>	The engine <code>d1.conf</code> file.

Configure the Engine to Use a Web Proxy

Configure the engine to use a web proxy, by adding the following configuration to the `d1.conf` file, with your specific proxy details:

```
{ "http_proxy": "http://proxy.host.local:8080",  
  "https_proxy": "https://proxy.host.local:8443" }
```

Configure the Engine to Call the Server Without Using a Proxy

In some cases, you may need to configure the engine to call the server without using a proxy when working with integrations (but not use a proxy when calling the Cortex XSOAR server).

STEP 1 | On the computer where you have installed the engine, go to the directory for `d1.conf` file.

For RPM, DEB, Shell go to `/usr/local/demisto`.

STEP 2 | Add the following configuration:

Key	Value
<code>engine.to.server.proxy</code>	<code>false</code> (default is <code>true</code>)

Configure the Number of Workers for the Server and Engine

If you experience performance issues, check the `workers.log` file to check if all workers are busy. Before increasing worker count, verify that there is enough CPU to handle the additional workers. You can configure the number of workers both for the server and engine or separately.

STEP 1 | To configure the number of workers for the Server, do the following:

1. Select **Settings > About > Troubleshooting > Add Server Configuration**.
2. Add the following key and value:

Task	Value
<code>workers.count.Tasks</code>	Number of workers for the server. Default: 4 workers per CPU core.

STEP 2 | To configure the number of workers for the engine, do the following:

1. Go to **Settings > Integrations > Engines**.
2. Select the engine checkbox you want to define the number of workers.
3. Click **Edit Configuration**.
4. In the **JSON formatted configuration** field, add the following engine configuration in JSON format:

Parameter	Value
<code>workers.count.engines</code>	Defines the number of workers for all engines across the system. This will override any other engine-worker configurations. Default is 4 workers per CPU core.
<code>workers.per.cpu</code>	Defines the number of workers per engine CPU. By default, each CPU has 4 workers, meaning that for an engine machine with 20 CPU, there will be 80 workers.

Configure Access to Communication Tasks through an Engine

You can use engines to enable users who do not have access to the Cortex XSOAR server to access the forms sent out in communication tasks. In such instances, the engine serves as a proxy and passes the information in the forms to the Cortex XSOAR server.

Prerequisites

Ensure that the **BindAddress** in the engine `d1.conf` file is configured to a port on which the engine can listen. The engine server address and the configured **BindAddress** are required to enable external users to communicate with the Cortex XSOAR server.

STEP 1 | Navigate to **Settings > About > Troubleshooting**.

STEP 2 | Click **Add Server Configuration**.

1. Under **Key**, enter `data.collection.external.link` and under **Value** enter the address of the engine through which the external users communicate.
2. Click **Save**.
3. Repeat this process for the `condition.ask.external.link` server configuration.

Notify Users When an Engine Disconnects

You can specify which Cortex XSOAR users receive an email notification when an engine disconnects.

STEP 1 | Go to **Settings > About > Troubleshooting**.

STEP 2 | In the **Server Configuration** section, add the following key and value.

Key	Value
<code>engines.notification.users</code>	A comma-separated list of Cortex XSOAR users. For example, <code>user1,user2,user3</code>

Remove the Cortex XSOAR Server From the Load-Balancing Group

You may need to remove the Cortex XSOAR server from the load-balancing group when there are two or more engines in the load-balancing group, or if you use engines to access integrations that are inaccessible from the Cortex XSOAR server.

STEP 1 | Select **Settings > About > Troubleshooting > Add Server Configuration**.

STEP 2 | Add the following key and value.

Key	Value
<code>engine.group.add.server.to.group</code>	False

The value is not case sensitive.

Remove an Engine

You can remove an engine depending on your operating system.

Run the following commands according to your operating system:

Operating System	Command
RPM	Get the full package: <code>rpm -qa grep -i ^d1_*</code> Remove the package: <code>rpm -evv d1_ <package name></code>
DEB	Get the full package: <code>dpkg-query -W -f='\${Package}' d1_*</code> Remove the package: <code>dpkg --purge <package name></code>
SH	Remove an engine: <code>sudo <engine-file-path> -- -purge</code>

Troubleshoot Cortex XSOAR Engines

When troubleshooting Cortex XSOAR engines, you need to access the logs from **Settings > Integrations > Engines** and select the engine you want to download the logs.

Debug Engines

The **d1.log** field appears whenever a Cortex XSOAR engine is running. The **d1.log** field contains information necessary for your customer success team to debug any engine related issue. The field displays any error, as well as noting whether the engine is connected.

```
1119 Sep 13 15:25 d1.cert.pem
685 Nov 29 1979 d1.conf
1679 Sep 13 15:25 d1.key.pem
477 Sep 13 15:25 d1.log
60623824 Sep 12 18:35 d1_darwin_amd64
61154501 Sep 12 18:35 d1_linux_amd64
61245440 Sep 12 18:35 d1_windows_amd64.exe
```

Engine Errors

These are some common Cortex XSOAR engine errors.

443 Error

This error might occur when a connection is established between an engine and the Cortex XSOAR server, because, by default, Linux does not allow processes to listen on low-level ports.

Error Message

```
listen tcp :443: bind: permission denied
```

Solution

- In the `d1.conf` file, change the port number to a higher one, for example, 8443.
- Run this command: `s**udo setcap CAP_NET_BIND_SERVICE=+eip /path/to/binary`. After running this command the server should be able to bind to low-numbered ports.

Troubleshoot Engine Upgrades

If the engine upgrade fails, you can manually upgrade the engine.

STEP 1 | Access the engine machine using a tool like SSH or PuTTY.

STEP 2 | Navigate to `/usr/local/demisto` and check which upgrade file exists.

File	Action
<code>d1_upgrade.sh</code>	<ol style="list-style-type: none">1. Run the <code>chmod +x /usr/local/demisto/d1_upgrade.sh</code> command.2. Run the <code>sudo /usr/local/demisto/d1_upgrade.sh</code> command.
<code>d1_upgrade_archive.sh</code>	<ol style="list-style-type: none">1. Run the <code>chmod +x /usr/local/demisto/d1_upgrade_archive.sh</code> command.2. Run the <code>sudo /usr/local/demisto/d1_upgrade_archive.sh</code> command.

Docker

- > [Docker Installation](#)
- > [Install Docker Images Offline](#)
- > [Configure Python Docker Integrations to Trust Custom Certificates](#)
- > [Docker Images in Cortex XSOAR](#)
- > [Docker Hardening Guide](#)
- > [Run Docker with Non-Root Internal Users](#)
- > [Use a Docker Image for Python Scripts](#)
- > [Troubleshoot Docker Networking Issues](#)

Docker Installation

Docker is used to run Python scripts and integrations in a controlled environment.

You can install Docker on the following Enterprise Linux platforms:

- [Docker Enterprise Edition](#)
- [Docker Community Edition](#)
- [Red Hat Docker Distribution](#)

Troubleshooting

- In some cases, the Docker created veths are not correctly bridged and the Docker container can't access the network or internet. You should update `systemd`.
- To verify that the Cortex XSOAR OS user has necessary permissions and can run Docker containers, run the following command from the OS command line.

```
sudo -u demisto docker run --rm -it demisto/python:1.3-alpine python --version
```

If everything is configured properly you will receive the following output. `Python 2.7.14.`

After installing Docker, you may need to configure [docker images](#) and update [settings](#).

Install Docker Enterprise Edition on Cortex XSOAR

Follow these steps to install Docker Enterprise Edition and Cortex XSOAR on RHEL7.x

STEP 1 | Install [Docker EE](#).

STEP 2 | Install [Cortex XSOAR](#).

STEP 3 | After Cortex XSOAR is installed, run the following commands:

```
sudo groupadd docker
sudo usermod -aG docker demisto
```

STEP 4 | Start the Docker service.

STEP 5 | (Optional) If you installed Cortex XSOAR before Docker EE, you should perform the following procedure.

You will receive an error message during the Cortex XSOAR installation. Acknowledge the error and then proceed.

1. Stop the Cortex XSOAR server.
2. Install Docker EE.
3. Run the following commands:

```
sudo groupadd docker
sudo usermod -aG docker demisto
```

4. Start the Cortex XSOAR server.
5. Select **Settings** > **About** > **Troubleshooting** > **Add Server Configuration**.
6. Remove the following keys:

```
python.executable
```

-
- ```
python.executable.no.docker
```
7. Start the Docker service.

## Install Docker Community Edition on Cortex XSOAR

It is recommended that you install Docker Enterprise Edition (EE), but it is still possible to install Docker Community Edition (CE). These instructions assume a clean Enterprise Linux installation without Docker EE.

### Before you begin:

- Ensure that you have the recommended storage drivers for **devicemapper** and **vfs**.
- Check the kernel version.

You must have Linux kernel 3.10 or above. To check your kernel version, run the following command:

```
$ uname -r
4.1.12-124.24.1.el7uek.x86_64
```

- [Update Container-Selinux](#) if you receive the **Requires: container-selinux >= 2.9** message.

**STEP 1** | Run the following commands:

- `sudo yum install -y yum-utils`
- `sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo`
- `sudo yum install -y --setopt=obsoletes=0 --nobest docker-ce`

**STEP 2** | After installation, start Docker daemon to fetch images during installation by running the following command:

```
systemctl start docker
```

**STEP 3** | Install Cortex XSOAR.

**STEP 4** | (Optional) If you are using Docker for an Engine, [Install an Engine](#).

**STEP 5** | Run the following commands:

- `sudo usermod -aG docker demisto`
- `systemctl restart docker`

## Update Container-Selinux

When installing Docker, if you receive the message **Requires: container-selinux >= 2.9**, you need to install a newer version of container-selinux.

**STEP 1** | Go to [CentOS Packages](#).

**STEP 2** | Find the latest version of `container-selinux` and copy the URL package.

**STEP 3** | Run the following command:

```
sudo yum install -y <copied container-selinux url
```

**STEP 4** | Install the latest version by running the following command (assuming the latest version is 2.74-1):

---

```
sudo yum install -y http://mirror.centos.org/centos/7/extras/x86_64/
Packages/container-selinux-2.74-1.el7.noarch.rpm
```

## Install Docker Distribution for Red Hat on Cortex XSOAR

Red Hat maintains its own package of Docker, which is the version used in OpenShift Container Platform environments, and is available in the RHEL Extras repository. For more information about the different packages available to install on Red Hat, see [Red Hat Knowledge Base Article](#) (requires a Red Hat subscription to access).

 RHEL 8 and later do not support this Docker installation method.

 CentOS 7 provides a similar `docker` distribution package as part of the CentOS Extras repository.

**STEP 1** | Install [Red Hat's Docker package](#).

**STEP 2** | Run the following commands:

```
systemctl enable docker.service
systemctl restart docker.service
```

**STEP 3** | Change ownership of the Docker daemon socket so members of the `dockerroot` user group have access.

1. Edit or create the file `/etc/docker/daemon.json`
2. Enable OS group `dockerroot` access to Docker by adding the following entry to the `/etc/docker/daemon.json`: `"group": "dockerroot"` file. For example:

```
{
 "group": "dockerroot"
}
```

3. Restart the Docker service by running the following command:

```
systemctl restart docker.service
```

4. Install [Cortex XSOAR](#).
5. After Cortex XSOAR is installed, run the following command to add the `demisto` os user to the `dockerroot` os group (Red Hat uses `dockerroot` group instead of `docker`):

```
usermod -aG dockerroot demisto
```

6. Restart the Cortex XSOAR server.

**STEP 4** | Set the required SELinux permissions:

Cortex XSOAR uses the `/var/lib/demisto/temp` directory (with subdirs) to copy files and receive files from running Docker containers. By default, when SELinux is in **enforcing** mode directories under `/var/lib/` it cannot be accessed by docker containers.

1. To allow containers access to the `/var/lib/demisto/temp` directory, you need to set the correct SELinux policy type by typing the following command:

```
chcon -Rt svirt_sandbox_file_t /var/lib/demisto/temp
```

2. (**Optional**) Verify that the directory has the `container_file_t` SELinux type attached by running the following command:

---

```
ls -d -Z /var/lib/demisto/temp.
```

---

# Install Docker Images Offline

Follow these steps to install Docker images when the Cortex XSOAR server is not connected to the internet.

**STEP 1** | Download the Docker image by appending the download link you received from Cortex XSOAR with the following parameters.

```
&downloadName=dockerimages
```

**STEP 2** | Copy the downloaded Docker image to the Cortex XSOAR server.

**STEP 3** | Stop the Cortex XSOAR service using the appropriate command for your OS.

- `systemctl stop demisto`
- `sudo service demisto stop`

**STEP 4** | Load the Docker image using the following command.

```
sudo docker load -i <YOUR_DOCKER_FILE>.tar
```

**STEP 5** | Start the Cortex XSOAR service using the appropriate command for your OS.

- `systemctl start demisto`
- `sudo service demisto start`

---

# Configure Python Docker Integrations to Trust Custom Certificates

Python integrations running in Docker contain a built-in set of CA-Signed certificates, to which you can add custom trusted certificates when needed. For example, if you are working with a proxy that performs SSL traffic inspection or using a service that has a self-signed certificate.



*Only PEM format certificates are supported.*

## STEP 1 | Configure the custom certificates.

1. Create a certificates PEM file that includes all of the required custom certificates.
2. (Optional) If you require the standard set of certificates trusted by browsers, download the [PEM certificates file](#) provided by the Certifi Project and add your custom certificates to the file that contains the standard set of certificates.

This example adds the `proxy-ca.pem` file (custom certificate) to the `cacert.pem` file (standard certificates): `cat proxy-ca.pem >> cacert.pem`

3. Copy the certificates PEM file to the following path.

```
/var/lib/demisto/python-ssl-certs.pem
```

## STEP 2 | Configure the Cortex XSOAR server settings.

1. Go to **Settings > About > Troubleshooting**.
2. In the **Server Configuration** section click **Add Server Configuration**.
  - Key: `python.docker.use_custom_certs`
  - `true`
3. Save the server configuration.
4. Restart the Cortex XSOAR server to verify that all existing Docker images are relaunched.

## STEP 3 | (Optional) Add the certificate files to engines.

1. Configure each engine to use the `/var/lib/demisto/python-ssl-certs.pem` file.
2. Make sure that you have the following directory on the engine host.

```
/var/lib/demisto
```

3. Set the `demisto` user as the directory owner with `0700` permissions.
4. Copy the `python-ssl-certs.pem` file to the `/var/lib/demisto` directory.
5. Add the following configuration to either the engine configuration file (UI) or to the `d1.conf` file.

```
"python.docker.use_custom_certs": true
```

6. Restart the engine.

## STEP 4 | Verify that the configuration was added successfully.

If you are using an SSL inspection proxy (MiTM) and want to verify that the certificates are properly set, you can run the following command, which will fetch from `www.google.com` using HTTPS, and print the headers of the response: `!py script="import requests; print(requests.get('https://google.com').headers)".`

After you save the server configuration, Docker images that are launched by the Cortex XSOAR server will contain the certificates file mounted in the following path:

---

```
/etc/custom-python-ssl/certs.pem
```

Additionally, the following environment variables will be set with the value of the certificates file path, which enables standard Python HTTP libraries to automatically trust the certificates (without code modifications):

- `REQUESTS_CA_BUNDLE`
- `SSL_CERT_FILE`

If you are developing your own integration (BYOI) and using non-standard HTTP libraries, you might need to include specific code that will trust the passed certificates file when the environment variable `SSL_CERT_FILE` is set. In these cases, always use the value in the environment variable as the path for the certificates file, and do not hard code the mounted path specified above. For example:

```
certs_file = os.environ.get('SSL_CERT_FILE')
if certs_file:
 # perform custom logic to trust certificates...
```

The [Python SSL library](#) will check the `SSL_CERT_FILE` environment variable only when using OpenSSL. If you are using a Docker image that uses LibreSSL, the `SSL_CERT_FILE` environment variable will be ignored.

---

# Docker Images in Cortex XSOAR

Docker is a tool used by developers to package dependencies into a single container (or image). This means that when creating an integration in Cortex XSOAR you are not required to “pip install” all required packages. The dependencies are part of a container that “docks” to the server and contains all libraries needed to run the integration. For more information see [Docker documentation](#).

## Why use Docker?

Docker is used to run Python scripts and integrations in a controlled environment. Integrations are run isolated from the server, which prevents accidental damage to the server. By packaging libraries and dependencies together, unknown issues can be prevented from occurring since the environments remain the same.

## Script and Integration Configuration

Specifying which Docker image to use is done in the Cortex XSOAR IDE (Open: **Settings** > **Docker image name**). If an image is not specified, a default Docker image using Python 2.7 is used. New scripts and integrations use Python 3, unless there is a specific reason not to use it. For example, a need to use a library which is not available for Python 3).



*You can specify in the Cortex XSOAR IDE the Python version (2.7 or 3.x). If 3.x is chosen, the latest Cortex XSOAR Python 3 Docker image is selected automatically.*

The selected Docker image is configured in the script or integration YAML file under the **dockerimage** key.

## Docker Images

Cortex XSOAR maintains a repository of Docker images, all of which are available in the [Docker hub under the demisto organization](#). The Docker image creation process is managed in the open-source project [demisto/dockerfiles](#). A search of the [repository-info branch](#) should be done prior to creating a new image. The repository is updated nightly with all image metadata and os/python packages used in the images.



*For security, images that are not part of the Cortex XSOAR organization in Docker hub cannot be accepted.*

When an engine needs a Docker image it pulls it either from Docker Hub or from a custom registry, if defined in the server configuration: **python.docker.registry**.

From version 5.0, the engine can fetch Docker images directly from the Cortex XSOAR server. If the engine fails to fetch the Docker image from the registry it tries to fetch it from the Cortex XSOAR server. The server packages the image when running **docker save**, and sends it to the engine, which enables the engine to obtain the required images, even if it does not have network access to the Docker Hub. The engine can only obtain images that are available from the server.

If an existing image cannot be found, you can [create a Docker image](#).

## Package Requirements

Consider some of the following:

- Does the package have known security issues?
- Is the package licensed?
- What type of license is used?

## Licensing

The Cortex XSOAR Content repository is only compatible with packages that use the MIT license. As a general rule, only use **permissive** licenses. For a complete list of licenses and types, see [comparison of free and open source software licenses](#).



*Other licenses might be permitted with specific approval.*

## Security Concerns

Due diligence needs to be done on all approved packages. Including verifying the package name is correct. In 2018 a scan of PyPI resulted in the detection of 11 “typo-squatted” packages which were found to be malicious. See [Detecting Cyber Attacks in the Python Package Index \(PyPI\)](#).

## Create a Docker Image in Cortex XSOAR

After due diligence has been completed and licenses checked, you can [Create a Docker Image In Cortex XSOAR](#).

### Docker Files (Required for Production)

If the integration is for public release, the integration pushes Docker files into the [dockerfiles repository](#). Pushing into the repository will add an image (after the approval process) to the Docker hub Cortex XSOAR organization. For more information, see [Cortex XSOAR's Dockerfiles and Image Build Management](#).



*When modifying an existing Docker image, ensure the change does not disrupt other integrations that may use the same package. All Docker images are created with unique version tags, for which overriding is blocked.*

## Manage Docker Images

You can find, create, and update Docker images using the following CLI commands. You can also configure the server to change the base Docker image, and define a Docker registry other than Docker hub.

### Display, Create and Update Docker Images

| Command                           | Description                                                                                                                                                                    |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>/docker_images</code>       | Displays all available Docker images.                                                                                                                                          |
| <code>/docker_image_create</code> | <a href="#">Create a Docker Image In Cortex XSOAR</a> .                                                                                                                        |
| <code>/docker_image_update</code> | Updates a specified, or all Docker images. Use this when you change a Docker image, and that image is used in a script, avoiding the need to manually update the Docker image. |

### Change the Base Docker Image

You need to add the following server configuration in **Settings > About > Troubleshooting**.

| Key                              | Value                                                                                                                                                           |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>python.docker.image</code> | The Docker image you want to define, as the base image. For example:<br><code>myregistry.local:5000/demisto/python:1.0</code><br><code>python:1.4-alpine</code> |

## Define a Docker Registry

You need to add the following server configuration in **Settings > About > Troubleshooting**

| Key                                 | Value                                                                                   |
|-------------------------------------|-----------------------------------------------------------------------------------------|
| <code>python.docker.registry</code> | The registry you want to point to. For example:<br><code>myregistry.local:5000</code> . |

If the alternate registry requires authentication you will need to login into the registry with the Cortex XSOAR OS user. Type the following:

```
sudo -u demisto docker login >registry server>
```

For more information about Docker login, see the [Docker documentation](#).

## Create a Docker Image In Cortex XSOAR

After due diligence has been completed and licenses checked, the following steps can be taken.

**STEP 1 |** In the War Room, type the following command:

```
/docker_image_create name
```

**STEP 2 |** Add the following arguments:

| Argument                  | Description                                                                                                                                 |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <code>name</code>         | New Docker image name. Lower case only                                                                                                      |
| <code>dependencies</code> | New Docker image dependencies. Python libs like stix or requests, can have multiple libs as comma separated: <code>lib1,lib2,lib3</code> .  |
| <code>packages</code>     | New Docker image packages. OS packages like libxslt or wget, can have multiple comma separated packages: <code>pkg1,pkg2,pkg3</code> .      |
| <code>base</code>         | New docker image base image to use. Must be ubuntu based with python installed. Default is demisto/python3-deb base image, with python 3.x. |

In the following example create a Docker image called `example_name` and use the python dependency, `Mechanize`. You can specify OS packages. This example requires `wget` as a package.

```
/docker_image_create name=example_name dependencies=mechanize packages=wget
```

When the Docker image is created, the following dialog box appears.

**Available Docker Images** ×

| ID           | Repository   | Size  | Tag    | Created Time                  | Created Since          |
|--------------|--------------|-------|--------|-------------------------------|------------------------|
| cc7288dfce3a | example_name | 717MB | latest | 2019-01-10 12:27:19 +0200 IST | Less than a second ago |

The Docker image is ready to use.

**STEP 3 |** (Optional) If you need to update a Docker image, type the following command:

---

`/docker_image_update`

**STEP 4** | Add the following arguments:

| Argument           | Description            |
|--------------------|------------------------|
| <code>image</code> | The name of the image. |
| <code>all</code>   | Pulls all images.      |

**STEP 5** | (Optional) To see all available images, type the following:

`/docker_images`

This command does not accept any arguments and lists all available Docker images.

---

# Docker Hardening Guide

This guide describes the recommended Cortex XSOAR settings for securely running Docker containers.

## Run Docker with a Non-Root Internal User

Running Docker containers with non-root internal users provides added security isolation and follows the principle of least privilege. For more information, see [Run Docker with Non-Root Internal Users](#).

## Limit Container Resources

To protect the host from a container using too many system resources (either because of a software bug or a DoS attack), limit the resources available for each container. In Cortex XSOAR, some of these settings are set using the advanced parameter: `python.pass.extra.keys`. This key receives as a parameter full `docker run` options, separated with the `##` string.

## Limit Available Memory

It is recommended limiting available memory for each container to 1 GB.



*On RHEL and CentOS 7.x distributions with Docker CE or EE with version 17.06 and later, ensure that your kernel fully supports `kmem` accounting or that it has been compiled to disable `kmem` accounting. The `kmem` accounting feature in Red Hat's Linux kernel has been reported to contain bugs, which cause kernel deadlock or slow kernel memory leaks. This is caused by a patch introduced in `runc`, which turns on `kmem` accounting automatically when user memory limitation is configured, even if not requested by the Docker CLI setting `--kernel-memory` (see: [opencontainers/runc#1350](#)). Users using Red Hat's distribution of Docker based on version 1.13.1 are not affected as this distribution of Docker does not include the `runc` patch. For more information see [Red Hat's Docker distribution documentation](#).*

*If you do not want to apply Docker memory limitations, due to the note above, you should explicitly set the advanced parameter: `limit.docker.memory` to `false`.*

**Swap Limit Support:** Not all Linux distributions have the swap limit support enabled by default.

- Red Hat and CentOS distributions usually have swap limit support enabled by default.
- Debian and Ubuntu distributions usually have swap limit support disabled by default.

To check if your system supports swap limit capabilities, after logging into the Server machine console (ssh), run the following command:

```
sudo docker run --rm -it --memory=1g demisto/python:1.3-alpine true
```

command. If you see the **WARNING: Your kernel does not support swap limit capabilities or the cgroup is not mounted. Memory limited without swap.** message in the output (the message may vary between Docker versions), you have two options:

- Configure `swap limit capabilities` by following the [Docker documentation](#).
- [Configure Memory Limit Support Without Swap Limit Capabilities](#).

If `swap limit capabilities` is enabled, [Configure the Memory Limitation](#).

To test the memory, see [Test the Memory Limit](#).

## Limit Available CPU

It is recommended limiting each container to 1 CPU. See [Limit Available CPU](#).

## Limit PIDs

It is recommend limiting each container to 256 PIDs. This value is sufficient for using threads and sub-processes, and protects against a fork bomb. You need to [configure the PIDs limit](#), by setting the `python.pass.extra.keys` advanced parameter. If you have this key already set up with a value append to it the config after a `##` separator.

### Limit Open File Descriptors

It is recommend using a soft/hard limit of 1024/8192 filed descriptors for each container process. You need to [Configure the Open File Descriptors Limit](#). If you have this key already setup with a value, append to it the config after a `##` separator.

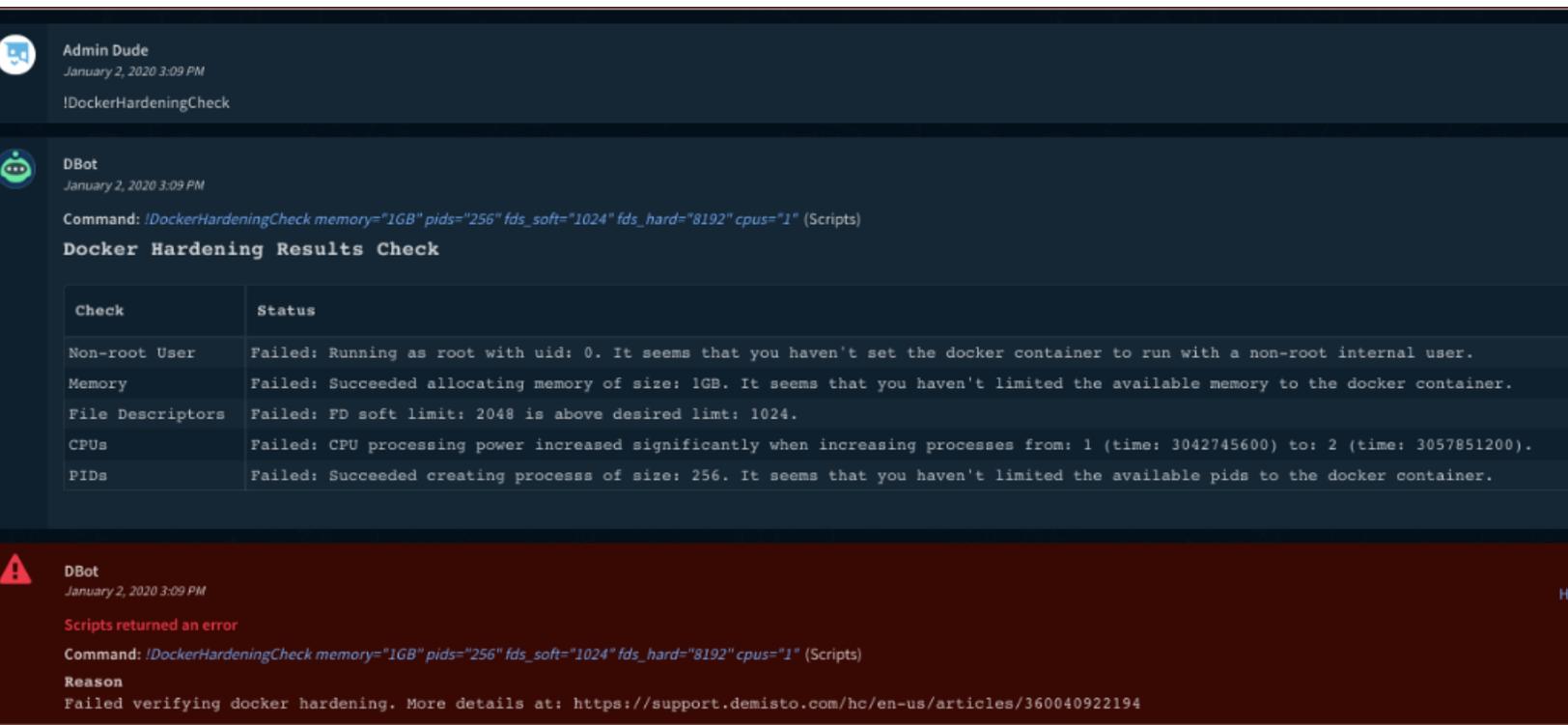
### Check Docker Hardening Configuration

To test that the hardening configuration has been applied correctly use the `DockerHardeningCheck` automation. For example:

#### Successful output:



#### Unsuccessful output:



### Working with engines

For each engine to which you want to apply Docker hardening, you need to [edit the engines' configuration](#) to include the Docker hardening advanced parameters. For example, you would add the following snippet to the configuration JSON file:

```
{ "docker.run.internal.asuser": true, "limit.docker.cpu": true, "limit.docker.memory": true, "python.pass.extra.keys": "--pids-limit=256##--ulimit=nofile=1024:8192" }
```

## Configure Memory Limit Support Without Swap Limit Capabilities

If you see the **WARNING: No swap limit support** you can configure memory support without **swap limit capabilities**.

**STEP 1 |** Set the `docker run` option `--memory-swap` option to `-1` (disables swap memory enforcement).

**STEP 2 |** In Cortex XSOAR, select **Settings > About > Troubleshooting > Add Server Configuration**.

**STEP 3 |** Type the following:

| Name                                | Value                                      |
|-------------------------------------|--------------------------------------------|
| <code>python.pass.extra.keys</code> | <code>--memory=1g##--memory-swap=-1</code> |

If you have the `python.pass.extra.keys` already set up with a value, append it to the config file are the `##` separator.

## Run Docker with Non-Root Internal Users

For additional security isolation, we recommend running Docker containers as non-root internal users. This follows the principle of least privilege.

**STEP 1 |** Configure Cortex XSOAR Server to execute containers as non-root internal users.

1. Select **Settings > About > Troubleshooting > Add Server Configuration**.
2. Add the following:

| Key                                     | Value             |
|-----------------------------------------|-------------------|
| <code>docker.run.internal.asuser</code> | <code>true</code> |

3. Click **Save**.
4. Reset the running containers using on of the following methods:  
From the Cortex XSOAR CLI, type `/reset_containers` command.  
Alternatively, restart the Cortex XSOAR Server.
5. From the Cortex XSOAR CLI, type the following command to check if the container is running as non-root internal user:

```
!py script="import os;print(os.getuid())"
```

If the server configuration was added successfully and the container is running with a non-root internal user, the output is a non-zero UID.



DBot May 17, 2019 1:27 PM

Command: `!py script="import os;print(os.getuid())"` (Scripts)

999

If the server configuration was not configured correctly and the container is running with an internal root user, the output is 0.

**STEP 2 |** For containers that do not support non-root internal users.

1. Select **Settings > About > Troubleshooting > Add Server Configuration**.
2. Add the following:

| Key                                     | Value                                                                                                                       |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <code>docker.run.internal.asuser</code> | A list of container names. The Cortex XSOAR server matches the container names according to the prefixes of the key values. |

For example, `docker.run.internal.asuser.ignore=demisto/python3:,demisto/python:`

The Cortex XSOAR server matches the key values for the following containers:

- demisto/python:1.3-alpine
- demisto/python:2.7.16.373
- demisto/python3:3.7.3.928
- demisto/python3:3.7.4.977

The `:` character should be used to limit the match to the full name of the container. For example, using the `:` character does not find `demisto/python-deb:2.7.16.373`.

## Use a Docker Image for Python Scripts

By default, Cortex XSOAR runs Python scripts in a Docker image. The default Docker image that Demisto uses is `demisto/python`, but you can use other Docker images that you have on your system. Cortex XSOAR supports the following Python versions:

- 2.7
- 3.0 and later

To disable Docker, contact Cortex XSOAR support.

**STEP 1 |** Access an automation script and go to **Advanced > Docker image**.

**STEP 2 |** In BYOI, set the Docker image to use to run the integration by expanding **Script > Python**.

## Configure the Memory Limitation

If `swap limit capabilities` is enabled, in Cortex XSOAR configure the memory limitation using the following advanced parameters.

**STEP 1 |** Select **Settings > About > Troubleshooting > Add Server Configuration**.

STEP 2 | Type the following:

| Key                              | Value             |
|----------------------------------|-------------------|
| <code>limit.docker.memory</code> | <code>true</code> |
| <code>docker.memory.limit</code> | <code>1g</code>   |

STEP 3 | Reset the running containers by typing the following command:

```
/reset_containers
```

## Test the Memory Limit

After configuring the memory limitation to the recommend 1 GB, you can test the memory limit in the playground.

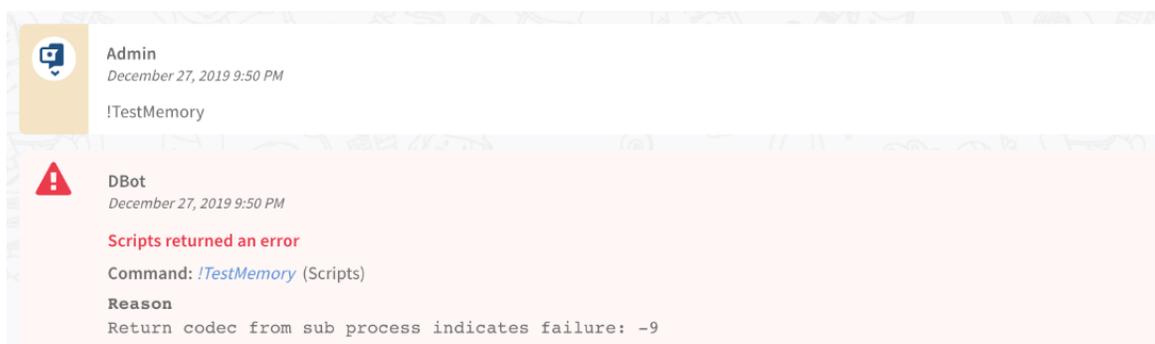
Run the following automation script:

```
from multiprocessing import Process
import os

def big_string(size):
 sys.stdin = os.fdopen(0, "r")
 s = 'a' * 1024
 while len(s) < size:
 s = s * 2
 print('completed creating string of length: {}'.format(len(s)))

size = 1 * 1024 * 1024 * 1024
p = Process(target=big_string, args=(size,))
p.start()
p.join()
if p.exitcode != 0:
 return_error("Return code from sub process indicates failure:
 {}".format(p.exitcode))
else:
 print("Success allocating memory of size: {}".format(size))
```

The command returns an error when it fails to allocate 1 GB of memory. For example:



---

## Limit Available CPU

Follow these instructions to set the advanced parameters to configure the CPU limit.

**STEP 1** | Select **Settings > About > Troubleshooting > Add Server Configuration**.

**STEP 2** | Type the following:

| Key                           | Value                                                        |
|-------------------------------|--------------------------------------------------------------|
| <code>limit.docker.cpu</code> | <code>true</code>                                            |
| <code>docker.cpu.limit</code> | <code>1.0</code> (Optional) Default is 1.0 if not specified. |

**STEP 3** | Reset the running containers by running the `/reset_containers` command.

## Configure the PIDs Limit

Configure the PIDs limit by setting the `python.pass.extra.keys` advanced parameter.

**STEP 1** | Select **Settings > About > Troubleshooting > Add Server Configuration**.

**STEP 2** | Type the following:

| Key                                 | Value                         |
|-------------------------------------|-------------------------------|
| <code>python.pass.extra.keys</code> | <code>--pids-limit=256</code> |

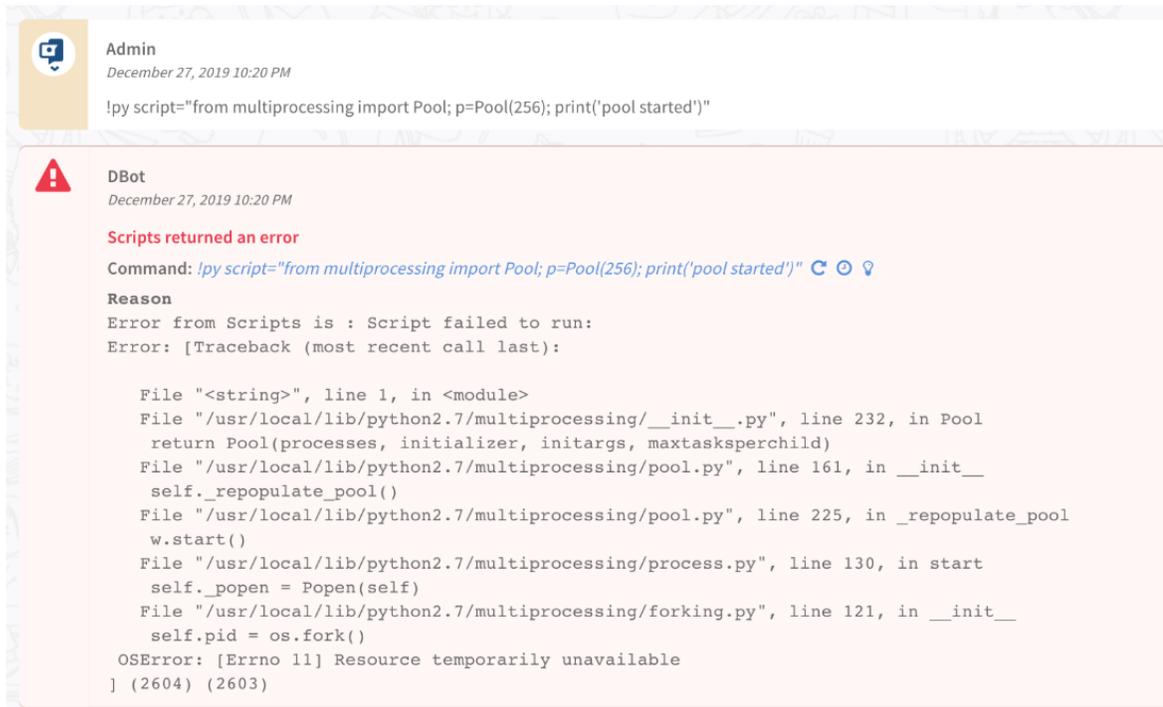
**STEP 3** | Reset the running containers by running the following command:

```
/reset_containers
```

**STEP 4** | (Optional) To Test the PIDs limit, run the type the following command in the playground:

```
!py script="from multiprocessing import Pool; p=Pool(256); print('pool started')"
```

When the limit is in place, the command fails with a Python OSErrror. For example:



## Configure the Open File Descriptors Limit

You need to set the `python.pass.extra.keys` advanced parameter to configure the open file descriptors limit.

**STEP 1** | Select **Settings > About > Troubleshooting > Add Server Configuration**.

**STEP 2** | Type the following:

| Key                                 | Value                                  |
|-------------------------------------|----------------------------------------|
| <code>python.pass.extra.keys</code> | <code>--ulimit=nofile=1024:8192</code> |

**STEP 3** | Reset the running containers by running the following command:

```
/reset_containers
```

**STEP 4** | (Optional) To test the file descriptors limit, run the following command in the playground:

```
!py script="import resource;print('file descriptor limit: ',
resource.getrlimit(resource.RLIMIT_NOFILE))"
```

The command prints the file descriptor limit (soft and hard). For example:



## Troubleshoot Docker Networking Issues

In Cortex XSOAR, integrations and automation scripts run either on the server or in a Docker container.

### Server-based integrations

Integrations and automation scripts that run on the server include native integrations (part of the server binary) and JavaScript integrations. JavaScript integrations run within the Cortex XSOAR server process using a JS virtual environment. These integrations use the same network IPs as the server.

### Docker-based integrations

These include integrations written in Python or Powershell. Docker creates its own networking, thus the integrations are using a different networking stack from the Cortex XSOAR server. The source IPs for these integrations are different and provided according to the Docker networking configuration.

When running integrations or automations that run within Docker containers you might encounter cases that networking fails for these integrations. The following are several examples of error messages that indicate a networking issue:

- [Errno -2] Name does not resolve
- [Errno 110] Operation timed out
- Failed to establish a new connection: [Errno -3] Try again

It is recommended that you use the [Docker networking stack](#) because it provides networking isolation and to consult the Docker documentation to resolve [Docker networking issues](#).

In cases that running with Docker's networking stack continues to cause issues, there is an option to run Docker containers with the host networking. In this mode, the container will share the host's network stack and all interfaces from the host will be available to the container. The container's host name will match the hostname on the host system. To enable host networking, add the following advanced server configuration in Cortex XSOAR:

- Key: `python.pass.extra.keys`
- Value: `--network=host`

After you add the server configuration, run the `/reset_containers` command from the Cortex XSOAR CLI to reset all containers and to begin using the new configuration.

### Notes

- For multi-tenant deployments, you need to add this setting to each tenant.
- When using engines, you need to add this setting to each engine.

# Run Docker with Non-Root Internal Users

For additional security isolation, we recommend running Docker containers as non-root internal users. This follows the principle of least privilege.

**STEP 1** | Configure Cortex XSOAR Server to execute containers as non-root internal users.

1. Select **Settings** > **About** > **Troubleshooting** > **Add Server Configuration**.
2. Add the following:

| Key                                     | Value             |
|-----------------------------------------|-------------------|
| <code>docker.run.internal.asuser</code> | <code>true</code> |

3. Click **Save**.
4. Reset the running containers using one of the following methods:  
From the Cortex XSOAR CLI, type `/reset_containers` command.  
Alternatively, restart the Cortex XSOAR Server.
5. From the Cortex XSOAR CLI, type the following command to check if the container is running as non-root internal user:

```
!py script="import os;print(os.getuid())"
```

If the server configuration was added successfully and the container is running with a non-root internal user, the output is a non-zero UID.



DBot May 17, 2019 1:27 PM

Command: `!py script="import os;print(os.getuid())"` (Scripts)

999

If the server configuration was not configured correctly and the container is running with an internal root user, the output is 0.

**STEP 2** | For containers that do not support non-root internal users.

1. Select **Settings** > **About** > **Troubleshooting** > **Add Server Configuration**.
2. Add the following:

| Key                                            | Value                                                                                                                       |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <code>docker.run.internal.asuser.ignore</code> | A list of container names. The Cortex XSOAR server matches the container names according to the prefixes of the key values. |

For example, `docker.run.internal.asuser.ignore=demisto/python3:,demisto/python:`

The Cortex XSOAR server matches the key values for the following containers:

demisto/python:1.3-alpine

demisto/python:2.7.16.373

---

demisto/python3:3.7.3.928

demisto/python3:3.7.4.977

The `:` character should be used to limit the match to the full name of the container. For example, using the `:` character does not find **demisto/python-deb:2.7.16.373**.



# ***Dashboards***

- > Dashboard Overview
- > Create a Dashboard
- > Add a Widget to a Dashboard
- > Configure a Default Dashboard
- > Share and Unshare a Dashboard
- > Edit a Dashboard



---

# Dashboard Overview

The dashboard consists of visualized data powered by fully customizable widgets, which enables you to analyze data from inside or outside Cortex XSOAR, in different formats such as graphs, pie charts, or text from information. For more information about widgets, see [Widgets Overview](#).

When you first install Cortex XSOAR, the following dashboard tabs are created:

- **Incidents:** information relating to incidents, such as severity type, active incidents, unassigned incidents and so on.
- **Threat intelligence Management:** information relating to threat intel management indicators.
- **System Health:** information relating to the Cortex XSOAR Server.
- **My Dashboard:** a personalized dashboard relating to your incidents, tasks, and so on.
- **SLA:** information relating to your Service Level Agreement.



*You can change the order of the dashboards in the dashboard tab by clicking next to the relevant dashboard, and then drag and drop the dashboard into the required location.*

In every dashboard, you can set the date range from which to return data and the refresh rate. In the **DASHBOARDS** tab, you can do the following:

- [Create a Dashboard](#)
- [Edit a Dashboard](#)
- Import and export a dashboard, which is useful in a test and production environment.
- [Share and Unshare a Dashboard](#)
- Delete or remove (if shared) a dashboard.

If you want to set up dashboards as a default for all existing and future users, see [Configure a Default Dashboard](#).

---

# Create a Dashboard

You can create and customize multiple dashboards, which are personal to the user.

**STEP 1** | From the homepage, in the **DASHBOARD** tab, click **New Dashboard**.

**STEP 2** | From the **Dashboards** page, in the **Widget's Library** section, [add the widgets to the dashboard](#).

**STEP 3** | From the **Date Range** drop down list, set the date range for the dashboard.

Widgets can have their own date range, which may be different than the dashboard's date range.

**STEP 4** | Type a name for the dashboard.

**STEP 5** | Click **Save** or **Save Version**.

Save Version enables you to view a history of the changes made to your dashboard. You can revert to previous versions.

---

# Add a Widget to a Dashboard

You can add a widget to an existing or a new dashboard.

**STEP 1** | [Create](#) or [edit a dashboard](#).

**STEP 2** | When editing a dashboard, click **Add Widget**.

**STEP 3** | In the **Widgets Library** section, search for the widget you want to add and click **Add**.

**STEP 4** | To edit the widget, select the **gear button** > **Edit Widget**.

**STEP 5** | Add the [Widget Parameters](#).

**STEP 6** | Click **Save**.

By default, a widget inherits the date range that you specify when creating the widget, but you can modify the date range when you create the dashboard or report. If the date range for the report or dashboard does not include the widget date range, the data is blank. To override the dashboard or report's date range, click **Use Widget's date range**.

---

# Configure a Default Dashboard

You can configure Cortex XSOAR to set a default dashboard for all existing and future users.

**STEP 1** | Go to **Settings > About > Troubleshooting**.

**STEP 2** | In the **Server Configuration** section, click **Add Server Configuration**.

**STEP 3** | Add the `default_dashboards` key with the dashboard ID value.

If left empty, Cortex XSOAR uses the out-of-the box dashboards as default. For more information, about out-of-the box dashboards, see [Dashboard Overview](#). To get the ID of the dashboard, export the dashboard, and open the JSON file.

**STEP 4** | Refresh the browser.

This procedure does not affect any custom dashboards already set by users.

---

# Share and Unshare a Dashboard

You can share dashboards with all users in your organization. Only the user that shares the dashboard has permission to stop sharing (unshare) the dashboard. You can share dashboards if you have changed a default dashboard or created a new one. Users have the option to add and remove the shared dashboard from their dashboard view.

**STEP 1** | Go to the dashboard you want to share/unshare.

**STEP 2** | To share a dashboard:

1. Select **gear button** > **Share**.
2. To add the shared dashboard, from the home page select **DASHBOARDS** > **Add dashboard** and select the dashboard you want to add.

**STEP 3** | To unshare a dashboard, from the drop down list, click **Unshare**.

**STEP 4** | To remove a shared dashboard, select **DASHBOARDS** > **gear icon** > **Remove**.

---

# Edit a Dashboard

You edit a dashboard by adding deleting or changing the widgets. You can also adjust the size and position of the widgets.

**STEP 1** | In the homepage, select the **gear button** > **Edit**.

**STEP 2** | Edit the widgets, as required:

- [Add a Widget to a Dashboard](#)
- [Edit a Widget](#)
- Remove a widget
- Change the **Date Range** for the dashboard
- Change the name of the dashboard

**STEP 3** | Click **Save** or **Save Version**.

Save Version enables you to view a history of the changes made to your dashboard. You can revert to previous versions. You can revert to the original default dashboards, by selecting **Reset Default Dashboards**.

# Reports

- > [Reports Overview](#)
- > [Chromium Installation for Reports](#)
- > [Configure Cortex XSOAR to Use PhantomJS](#)
- > [Create a Report](#)
- > [Schedule a report](#)
- > [Create an Incident Summary Report](#)
- > [Add a Widget to a Report](#)
- > [Edit a report](#)
- > [Change the Report Logo](#)
- > [Configure the Time Zone and Format in a Report](#)
- > [Troubleshoot Reports](#)



---

# Reports Overview

Reports contain statistical data information which enables you to analyze data in PDF, Word and CSV formats. A report contains [widgets](#) , which enables you to analyze data from inside or outside Cortex XSOAR, in different formats such as graphs, pie charts, or text from information.

Cortex XSOAR uses Chromium or Chrome to generate reports. If your operating system does not have Chrome or Chromium you need to [install](#). Alternatively, you can use PhantomJS (deprecated), which is not developer supported. You need to [Configure Cortex XSOAR to Use PhantomJS](#).

Reports can be time bound or non-time related. Time bound reports are incident summaries over a period of time. For example, incident summaries over the last 24 hours, or last 30 days. Non-time related reports are filtered summaries of incident data. For example, open, late, and critical incidents.

Cortex XSOAR comes with out-of-the-box reports, such as critical and High incidents, Daily incidents, last 7 days incidents, and so on. These reports cannot be edited apart from the schedule time and who can receive the report.



*If you want to change these type of reports, go to [github reports repository](#), download and update the JSON file, and upload the report.*

You can do the following:

- [Create a Report](#)
- [Add a Widget to a Report](#)
- [Edit a report](#)
- [Change the Report Logo](#)
- [Configure the Time Zone and Format in a Report](#)



*For custom reports you cannot generate CSV reports unless you define it in a JSON file and import it to Cortex XSOAR.*

You can run the report immediately or schedule a time as described in [Schedule a report](#).

You can schedule a report directly from incident, as described in [Create an Incident Summary Report](#).

---

# Chromium Installation for Reports

Cortex XSOAR uses Chromium or Chrome to generate reports. If Chromium or Chrome is not installed, the report will be generated using PhantomJS, which is deprecated and not developer supported. For more information about how to use PhantomJS, see [Configure Cortex XSOAR to Use PhantomJS](#).

By default, Chromium or Chrome is installed when installing Cortex XSOAR for most operating systems. However if you use one of the following operating systems, you need to download and install Chromium.

- [Fedora, RHEL, or CentOS](#)
- [openSUSE and SUSE](#)
- [Ubuntu or Debian](#)

## Install Chromium on Fedora, RHEL, or CentOS

If you are using Fedora, RHEL, or CentOS you need to download and install Chromium or Chrome for your operating system.

**STEP 1 |** Install the Chromium package from the repository, by typing `$ sudo sh -c 'echo -e "[google-chrome]\nname=google-chrome - 64-bit\nbaseurl=http://dl.google.com/linux/chrome/rpm/stable/x86_64\nenabled=1\nngpgcheck=1\nngpgkey=https://dl-ssl.google.com/linux/linux_signing_key.pub" >> /etc/yum.repos.d/google-chrome.repo'`.

**STEP 2 |** Refresh the repository, by typing `$ sudo yum update`.

**STEP 3 |** Install the stable version of Chromium by typing `$ yum install google-chrome-stable`.

## Install Chromium on openSUSE and SUSE

If you are using SUSE or openSUSE, download and install Chromium or Chrome for your operating system.

**STEP 1 |** Install the Chromium package from the repository by typing, `$ sudo zypper ar http://dl.google.com/linux/chrome/rpm/stable/i386 Google-Chrome`.

**STEP 2 |** Refresh the repository by typing, `$ sudo zypper refresh`.

**STEP 3 |** Install the stable version of Chromium by typing, `$ sudo zypper install google-chrome-stable`.

## Install Chromium on Ubuntu or Debian

If you are using Ubuntu or Debian, download and install Chromium or Chrome for your operating system.

**STEP 1 |** Install the Chromium package from the repository by typing, `$ sudo sh -c 'echo "deb http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/sources.list.d/google.list'`.

**STEP 2 |** Add the SSL key by typing, `$ sudo sh -c 'echo "deb http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/sources.list.d/google.list'`.

---

**STEP 3** | Refresh the repository by typing, `$ sudo sh -c 'echo "deb http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/sources.list.d/google.list'`.

**STEP 4** | Install the stable version of Chromium by typing, `$ sudo sh -c 'echo "deb http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/sources.list.d/google.list'`.

---

# Configure Cortex XSOAR to Use PhantomJS

Cortex XSOAR uses Chrome to generate Cortex XSOAR reports. You either need to install Chrome or use PhantomJS (deprecated) to generate reports. You need to add a server configuration to use PhantomJS.

**STEP 1** | Select **Settings > About > Troubleshooting**.

**STEP 2** | In the **Server Configuration** section, click **Add Server Configuration**.

**STEP 3** | In the **Key** field, type `reports.force.phantomjs`

**STEP 4** | In the **Value** field, type `true`

**STEP 5** | Click **Save**.

---

# Create a Report

You can create and customize reports in the **REPORTS** tab. You can add widgets, schedule times, add recipients, change the format, size, and so on.

Reports are generated using Chromium browsers. If you do not have a Chromium browser you can either download the browser or generate reports using PhantomJS (deprecated). For more information about how to use PhantomJS, see [Configure Cortex XSOAR to Use PhantomJS](#).

**STEP 1** | From the homepage, in the **REPORTS** tab, click **New Report**.

**STEP 2** | From the **Reports** page, in the **Widget's Library** section, add the widgets as required, as described in [Add a Widget to a Report](#).

**STEP 3** | From the **Date Range** drop down list, select the date range from which to generate the report.  
Widgets can have their own date range, which can be different from the report's date range

**STEP 4** | To schedule a report, see [Schedule a report](#).

**STEP 5** | To change the number of recipients or their details, in the **Recipients** field, click the number of recipients.

**STEP 6** | To change the format, orientation and paper size select the options as required.  
It is recommended to use landscape to ensure that all information displays in the report.

**STEP 7** | Before generating the report, click **Preview** to see a preview of the report. You can change the size or arrange the widgets as required,

**STEP 8** | Click **Save** or **Save Version**.

Save Version enables you to view a history of the changes made to your report. You can revert to previous reports.

**STEP 9** | To generate the report immediately, in the **Reports** tab, click **Run Now**.

The Report downloads.

Ensure that you enable pop-ups in your browser.

---

# Schedule a report

You can schedule a report to run on particular times with start and end dates. You can also add restrictions on the report content and the number of recipients. If you want to send the report to users by email, you need to add an email integration instance, such as EWS, Gmail, EWS, Mail Sender, and so on.

**STEP 1** | In the **Reports** tab, select the report you want to schedule.

**STEP 2** | In the **Next Run** field, click **Disabled** or the date it was last run.

If creating or editing a report, click next to the **Schedule** field.

**STEP 3** | In the dialog box, add the following information:

- The name of the recipients you want to send the report, if required.
- Select the **Scheduled** check box.

**STEP 4** | If you want to restrict the content of the report in accordance with a user's authorization, select the **Run with current user**.

To change authorizations, go to **Settings > USERS AND ROLES**. For more information about users and roles, see [Reports Overview](#).

**STEP 5** | Schedule a report according to one of the following methods:

- **Cron view**: Schedules a report according to a Cron time string format, which consists of five fields that Cron converts into a time interval. Use this view to schedule a report on certain hours, days, months, years, and so on. For examples of Cron strings, see [Schedule a Report Examples](#).  
When using the **Cron view** the **Start at** and **Ends** fields may conflict with the Cron string expressions.
- **Human view**: Schedules a report according to the set number of hours. You can add days of the week with start and end times.

When scheduling a report in the **Human view** the **Next Run** date may be incorrect. You may need to change the number of hours field when scheduling the report.

**STEP 6** | Click **Save**.

The schedule date appears in the **Next Run** field.

## Schedule a Report Examples

The following examples describe how to schedule a report using the using Cron scheduler format. The Cron time string format consists of five fields that Cron converts into a time interval. For example, a Cron string of `0 10 15 * *` runs a report on 15th of each month at 10:00.

### *Schedule a report starting January 1 and then monthly*

In this example, you want to schedule a report on January 1, 2020 at 0800 and thereafter 1st of each month.

In the **Cron Expression** field, type `00 8 1 1/1 *`

| Number | Description   |
|--------|---------------|
| 00     | 00 in minutes |

| Number | Description                                                                                                                                                                                 |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8      | 8am                                                                                                                                                                                         |
| 1      | The first of each month                                                                                                                                                                     |
| 1/1    | Starting in January, and every month thereafter. If you want the report to start on a different month, change 1/1 to the relevant month, such as 2/1 for February, 3/1 for March and so on. |
| *      | Any day of the week                                                                                                                                                                         |

The reports run at 8am on January 1, 2020, February 1, 2020, March 1, 2020 and so on.



*Cron calculates the next relevant date. If you want the report to run next month, provided that date has passed in the current month, you do not need to specify the month. For example, assume the date is 12 December. To run the report on 11 January at 8am, type 00 8 11 \* \*. The report starts running on 11 January (and on 11th of each month thereafter). If the current date is 10 December, the next run date would be 11 December.*

## Schedule a report on once a year

In this example you want to schedule a report on 6 January 2020 at 0800 and every year on the 1 January (the current date is Thursday 12 December 2019).

In the **Cron Expression** field, type `00 8 1 1 *`

| Number | Description                                                    |
|--------|----------------------------------------------------------------|
| 00     | 00 minutes                                                     |
| 8      | 8am                                                            |
| 1      | 1st day of each month                                          |
| 1      | Starting every January. For different months change the number |
| *      | Any day of the week                                            |

The report runs at 0800 on 1 January 2020, 1 January 2021, 1 January 2022, and so on.

## Schedule a report every week on a Monday

In this example, you need to schedule a report at midnight every week on a Monday (the current date is Thursday, 12 December 2019)

Type the following expression in Cron: `00 0 * * 1`

| Number | Description   |
|--------|---------------|
| 00     | 00 in minutes |
| 0      | Midnight      |

| Number | Description |
|--------|-------------|
| *      | Any day     |
| *      | Any month   |
| 1      | Monday      |

The report runs on the first available Monday 16 December at midnight, and on 23 December, 30 December, 6 January, and so on.

### *Schedule a report every weekday from February for 6 months*

In this example, you need to schedule a report at 5.30pm every weekday (Mon - Fri) starting in February for the 6 months (assume the current date is Thursday 12 December 2019).

In the **Cron Expression** field, type `30 17 * 2/6 1-5`

| Number | Description                                 |
|--------|---------------------------------------------|
| 30     | 30 minutes                                  |
| 17     | 5pm                                         |
| *      | Any day                                     |
| 2/6    | Starting in February for the next 6 months. |
| 1-5    | Monday to Friday                            |

The report runs at 5.30pm on February 3, 4, 5, 6, 7 and so on.

### *Schedule a daily report*

In this example, you need to schedule a report every day at 0600 (the current date is Wednesday 12 December).

In the **Cron Expression** field, type `0 6 * * *`

| Number | Description                                                                                                |
|--------|------------------------------------------------------------------------------------------------------------|
| 00     | 00 in minutes                                                                                              |
| 6      | 6am                                                                                                        |
| *      | Any day                                                                                                    |
| *      | Any month                                                                                                  |
| *      | Any day of the week. If you want to run from Monday to Friday, type 1-5. For Sunday to Thursday, type 0-4. |

The report runs at 6am on 13 December, 14, 15, 16 and so on.

---

# Create an Incident Summary Report

You can create reports based on the tabs in the **Incident** page, which enables you to capture investigation-specific data and share it with team members.

**STEP 1** | Go to the **Incidents** page and select the incident for which you want to create a report.

**STEP 2** | In the **Case Info** tab, select **Actions > Report**.

**STEP 3** | To build a new report, from the **Build Report** tab, select the following:

- **Format**
- **Orientation**

It is recommend to use the landscape orientation to ensure that all information displays in the report.

- **Paper Size**

If you want to use the setting as a template, click the **Save report as template** check box.

**STEP 4** | To use an existing template, from the **Select a Template** tab, select the template.

**STEP 5** | Click **Generate report**.

You can see a list of incident reports in the **Reports** page.

---

# Add a Widget to a Report

You can add a widget to an existing or a new dashboard.

**STEP 1** | Create or edit a report.

**STEP 2** | When editing a report, click **Add Widget**

**STEP 3** | In the **Widgets Library** section, search for the widget you want to add and click **Add**.

**STEP 4** | To edit the widget, click the gear button and select **Edit Widget**.

**STEP 5** | Add the [Widget Parameters](#).

**STEP 6** | Click **Save**.

By default, a widget inherits the date range that you specify when creating the widget. If the date range for the report or dashboard does not include the widget date range, the data is blank. To change the widget's date range, click the gear icon and select either **Use Widget's date range**, or **Use Dashboard's date range**.

---

# Edit a report

You can edit custom reports by adding, editing and removing widgets, changing the layout, output type and so on. You cannot edit out-of-the-box reports or reports that have been created from an incident summary.

**STEP 1** | In the **Reports** page, locate the report you want to edit and click the edit button.

**STEP 2** | Update the report as required:

- [Add a Widget to a Report](#)
- [Edit a Widget](#)
- Remove a widget
- Change the date range from the **Date Range** drop down list.
- [Schedule a report](#)
- Change the number of recipients or their details by clicking the number of recipients in the **Recipients** field.
- Change the format, orientation and paper size, as required.

---

# Change the Report Logo

By default, reports include the Cortex XSOAR logo. You can change the logo to match or your organization branding.

**STEP 1** | Save your logo as a base64 image or upload it to a URL.

**STEP 2** | Go to **Settings > About > Troubleshooting**.

**STEP 3** | In the **Server Configuration** section, click **Add Server Configuration**.

**STEP 4** | Add the following keys and values:

- Key: **reports.logo.customer**  
Value: *The base64 image or URL for your logo.*
- Key: **reports.logo.demisto**  
Value: **False**

---

# Configure the Time Zone and Format in a Report

You can set the time and date format, and a time zone for reports by adding a server configuration. When this is not specified, the time/date format and time zone are the local time and location when the report is generated.



*Most out of the box reports, time zone and time formats cannot be changed. For custom reports, custom fields can be changed.*

**STEP 1** | Go to **Settings > About > Troubleshooting**.

**STEP 2** | In the **Server Configuration** section, click **Add Server Configuration**.

**STEP 3** | Add the following keys and values:

| Key                              | Value                                                                                                                                                                                                                                                                              |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>reports.time.format</code> | The time/date for your report. For example: <ul style="list-style-type: none"><li>• 02 Jan 20 15:05 GMT</li><li>• Thursday, 02-Jan-20 15:05 MST</li><li>• 2020-01-02T15:05:10Z07:00</li><li>• 01-Jan-Month, 02-02-Day, 03-15-hours, 04-minutes, 05-seconds, 20-2020-Year</li></ul> |
| <code>reports.time.zone</code>   | The time zone for your report. For example: <ul style="list-style-type: none"><li>• Asia/Jerusalem</li><li>• UTC</li><li>• America/New York</li><li>• CET</li><li>• EST</li><li>• GMT</li></ul>                                                                                    |

**STEP 4** | Click **Save**.

---

# Troubleshoot Reports

By default, when you generate a report the JSON file is not retained. This JSON file is needed to troubleshoot report issues.

**STEP 1** | Go to **Settings > About > Troubleshooting**.

**STEP 2** | Click **Add Server Configuration**.

- Key: **report.remove.data**
- Value: **false**

**STEP 3** | Generate the report.

**STEP 4** | Download the JSON file from `/var/lib/demisto/temp/`.

**STEP 5** | Go to **Settings > About > Troubleshooting** and delete the server configuration you created.

# Widgets

- > [Widgets Overview](#)
- > [Create a Widget in the Widgets Library](#)
- > [Create a Custom Widget Using a JSON File](#)
- > [Create a Custom Widget Using an Automation Script](#)
- > [Edit a Widget](#)
- > [Create a Used Percentage Widget for a Disk Partition](#)
- > [Saved By Dbot \(ROI\) Widget](#)



---

# Widgets Overview

Widgets are visual components that enable you to analyze data internally or externally from Cortex XSOAR, in different formats such as graphs, pie charts, text from information, and so on.

Cortex XSOAR comes with a number of out of the box system widgets, such as **Today's New Incidents**, **Late Incidents**, and **Saved by Dbot**, etc. You can edit these widgets, when creating or editing a dashboard or report.



*Some out of the box system widgets are not editable. If you want to change these widgets, go to the [github widgets repository](#), download, update the JSON file, and upload it to the **Widgets Library**.*

You can create widgets from the following and then add them to a [dashboard](#) or [report](#), as required:

- **Widgets Library**: create the widget in the **Widgets Library** which is then available for all users.
- **Create a Widget From an Incident**: create the widget from the incident page and then add it to a dashboard or a report.
- **Create a Widget from an Indicator**: create the widget from the indicators page then add it to a dashboard or a report.
- **JSON file**: these are static widgets and display relatively straightforward information, such as grouping incidents severity by type, active incidents by type, and so on.
- **Automation Script**: you can create dynamic widgets using automation scripts for more complex calculations, such as calculating the percentage of incidents that DBot closed. The automation script can pull information from the Cortex XSOAR API. For examples, see [Script Based Widgets Using Automation Scripts Examples](#).

If you want to add a script based widget to a dashboard or report, you need to create a widget in the **Widgets Library**. You can create or upload the script to the **Automation** page or you can directly upload the script to the **Widgets Library**.

You can also [add a custom widget in the War Room](#), so you can easily view the incident in a widget format, such as severity in a bar chart.



*If you have a significant numbers of widgets, performance may be affected. You should try to keep widgets simple (no scripts) and refresh times higher than 1 minute whenever possible.*

---

# Create a Widget in the Widgets Library

Widgets are visual components that populate dashboards and reports with specific data. Although there are various out-of-the-box system widgets available, you can create custom widgets in the Widgets Library. You can also create them from an [incident](#) or an [indicator](#).

**STEP 1** | Create or edit a [report](#) or [dashboard](#).

**STEP 2** | ([Optional](#)) If editing an existing report or dashboard, click **Add Widget**.

**STEP 3** | In the **Widgets Library** click the **+** button.

**STEP 4** | From the drop down list, select one of the following data types:

- Incidents Data
- Indicators Data
- Script based
  - Relevant if you have created a script in the **Automation** page.
- Upload
  - You can upload either a JSON file or a script file.

You can change the data type when you edit the widget.

**STEP 5** | In the **Quick chart definitions** window, select the [Widget Parameters](#).

**STEP 6** | Click **Save**.

The widget is added to the widgets library.

**STEP 7** | Add the widget to the dashboard or report.

## Widget Parameters

The following table describes the widget parameters in the **Quick chart definitions** window.

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Type | <p>The type of data you want to display. From the drop down list, you can select the following:</p> <ul style="list-style-type: none"><li>• Incidents</li><li>• Indicators</li><li>• Scripts</li></ul> <p>When selecting Scripts, if your script does not appear you need to add it to the <b>Automation</b> page and add the <b>widget</b> label.</p> <p> <i>For some widgets you cannot select the data type, such as task widgets.</i></p> |

| Parameter                                                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data query                                                                          | Queries query data in the Lucene query syntax form relating to the dataType. For example when dataType is incidents and the query is: -status:closed and owner:"", it queries all incidents that are not closed, which does not have an owner.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Time frame                                                                          | The time frame to retrieve data.<br><br> <i>When you add the widget, it automatically uses the date range of the dashboard or report. You can change it by clicking the gear icon and selecting Use widget's date range. To revert, click the gear icon again and select Use dashboard's date range.</i>                                                                                                                                                                                                                                                                                                                        |
| Incidents by                                                                        | Filters the data according to the incident type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Indicators by                                                                       | Filters the data according to the indicator.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Tasks by                                                                            | Filters the data according to the Task ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Entries by                                                                          | Filter the data according to the entry, such as the date created.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|    | Views data in a bar format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|    | Views data in a column format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|  | Views data in a line graph format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|  | View the data in a pie format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|  | Views data in a number format. You can do the following: <ul style="list-style-type: none"> <li>To compare dates for a particular period, select the <b>Display Trend</b> check box. For example, this week vs. last week, this year vs. last year, and so on. To change the comparison period, in the <b>Time frame</b> field from the drop down list, select the relevant date.</li> <li>Select a widget color threshold to highlight the threshold data and define the threshold by selecting the <b>Widget color threshold</b> check box. For example, if less than 150 red, 100 yellow, 50 green. To add more thresholds, click <b>Add new threshold</b>. You can change the colors as required.</li> </ul> |
|  | Views data in a table format. Click the gear icon to edit columns.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|  | Views data in a text format, which can be used as a text summary of the displayed data. You can use {0} to display a query value and {date} to display the date. Markdown is supported.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

---

# Create a Custom Widget Using a JSON File

You can create a custom widget for your dashboard or report using a JSON file and then add the new widget to a new or edited dashboard or report. If you want to create more complicated widgets using scripts, see [Create a Custom Widget Using an Automation Script](#).

**STEP 1** | Create a JSON file, and add the [JSON File Widget Parameters](#).

**STEP 2** | Create or edit a dashboard or report.

**STEP 3** | In the **Widget Library** section, select the **add button** > **Upload**.

**STEP 4** | Select the JSON file you created in step **1** and click **Open**.

**STEP 5** | To add the widget to the dashboard or report, click **Add**.

## JSON File Widget Parameters

The following table describes the parameters for a JSON file when creating a widget. For an example of a JSON file, see [JSON File Widget Example](#).

| Parameter | Description                                                                                                                                                                                                                                                                                                         |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id        | The unique identifier for the widget.                                                                                                                                                                                                                                                                               |
| name      | The display name of the widget.                                                                                                                                                                                                                                                                                     |
| datatype  | The data source of the widget. Must be one of the following: <ul style="list-style-type: none"><li>incidents</li><li>indicators</li><li>messages</li><li>entries</li><li>scripts</li></ul> Relevant only when you are creating an automation script. <ul style="list-style-type: none"><li>tasks</li></ul>          |
| query     | Queries query data in the Lucene query syntax form relating to the dataType. For example when dataType is incidents and the query is: -status:closed and owner:"" it queries all incidents that are not closed, which does not have an owner.<br><br>For script based widgets, the query is the name of the script. |
| sort      | Sorts the data, when displaying the widgetType (such as table, list, bar, column, pie) according to the following: <ul style="list-style-type: none"><li>field: the field name for which to sort.</li><li>asc: whether to sort data in ascending values. If true, the order is in ascending value.</li></ul>        |

| Parameter   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| widgetType  | <p>The type of widget you want to create. Must be one of the following:</p> <ul style="list-style-type: none"> <li>• bar</li> <li>• column</li> <li>• pie</li> <li>• number</li> <li>• line</li> <li>• table</li> <li>• trend</li> <li>• list</li> <li>• duration</li> <li>• image</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| size        | <p>The maximum number of returning elements. Use 0 for the widgetType's default. Note the following:</p> <ul style="list-style-type: none"> <li>• <b>Table/List:</b> To change the size, go to <b>Settings &gt; About &gt; Troubleshooting &gt; Add Server Configuration</b> add the <b>default.statistics.table.size.</b> key and then add the value. Default is up to 13</li> <li>• <b>Chart:</b> Default is up to 10.</li> <li>• <b>Number and Trend:</b> Ignores the size value.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| category    | <p>Adds a category name. The widget appears under a category instead of being classified by dataType.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| dataRange   | <p>The time period for which to return data. The time period is overridden by the dashboard or report time period. Default is all times.</p> <ul style="list-style-type: none"> <li>• <b>fromDate:</b> The start date from which to return data in the format: "YYYY-MM-DDTHH:MM:SSZ". For example, "2019-01-01T16:30:00Z"</li> <li>• <b>toDate:</b> The end date for which to return data in the format: "YYYY-MM-DDTHH:MM:SSZ". For example, "2019-01-01T16:30:00Z" -</li> <li>• <b>period:</b> An object describing a period of relative time. If using the <b>fromDate/toDate</b> parameters, this parameter is ignored. <ul style="list-style-type: none"> <li>• <b>byTo:</b> The <i>to</i> period unit of measurement. Values are 'minutes', 'hours', 'days', 'weeks', 'months'.</li> <li>• <b>byFrom:</b> The <i>from</i> period unit of measurement. Values are: 'hours', 'days', 'weeks', 'months'.</li> <li>• <b>toValue:</b> The duration of the <i>to</i> period. Integer.</li> <li>• <b>fromValue:</b> The duration of the <i>from</i> period. Integer. For example, last 7 days - { byFrom: 'days', fromValue: 7 }.</li> </ul> </li> </ul> |
| description | <p>The description of the widget in the Widget Library.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| params      | <p>Enriches the widget with specific parameters, mainly based on the widgetType. Includes the following:</p> <ul style="list-style-type: none"> <li>• <b>groupBy:</b> An array of field names for which to group the returned values. Used when widget type is bar, column, line or pie. For example, ["type", "owner"], groups results by type and owner, and returns a nested result for each type with statistics according to owner.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           |  <i>Bar/column charts defined with two groups can become stacked.</i> <ul style="list-style-type: none"> <li>• <b>hideLegend:</b> Shows or hides the legend, if it exists. Default is false.</li> <li>• <b>keys:</b> An array that enables processing the data value and modifies it by the given list of keys. For example, [ "avg   openDuration / ( 3600*24 ) " ] process for each group found in the result, the average open duration (in days).</li> <li>• <b>text:</b> The markdown text for text widgets or image data for image widgets. For example, if you want the widgets to appear on separate pages in a report, use [ "\pagebreak" ].</li> <li>• <b>timeFrame:</b> Supplies the custom time frame for which the widget scales. Values are "years", "months", "days", "hours", "minutes". The default is "days".</li> <li>• <b>tableColumns:</b> Enables you to define the name of the columns in a list or table. For example, "[{ "key": "name" }, { "key": "mycustomfield" }]", displays the name and a custom field.</li> </ul> |
| legend    | An array of objects that consists of a name and color. The name must match a group name. The color can be the name of the color, the hexadecimal representation of the color, or the rgb color value. (V6.0+)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

## JSON File Widget Example

In the following example, create a JSON file to display Incidents severity by type, which contains the following:

- Bar chart
- Incidents from the last 30 days
- Grouped by severity and for each severity display the nested group size (count of incidents displayed by the length of the bar) colored according to type.

Create the following JSON file:

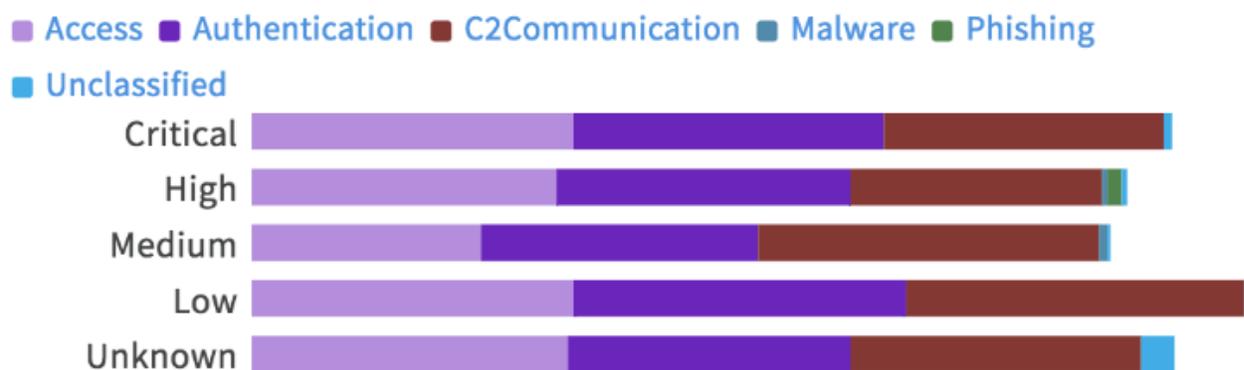
```
{
 "name": "Incident Severity by Type",
 "dataType": "incidents",
 "widgetType": "bar",
 "query": "-category:job and -status:archived and -status:closed",
 "dateRange": {
 "period": {
 "byFrom": "days",
 "fromValue": 30
 }
 },
 "params": {
 "groupBy": [
 "severity",
 "type"
]
 }
}
```

You can see the following parameters:

- The Widget is called **Incident Severity by type**.
- The data type is **incidents**.
- The widget type is **bar**.
- The **query** specifies that you do not want to return incidents that are categorized as job nor incidents that are archived and closed.
- For the date range, the **fromValue** sets the widget to display the last 30 units of time. The **byFrom** sets the units of time to days, which results in the last 30 days.
- The **params** parameter is set with a **groupBy** value marking the first group by severity name and then by type (making the bar chart stacked).

After you import the widget into the **Widget Library** the following widget appears:

## Incident Severity by Type



You can see the incidents are grouped by severity and the number of incidents are displayed by the length of the bar, which are colored according to type.

---

# Create a Custom Widget Using an Automation Script

You can use custom script based widgets with an automation script to create dynamic widgets for more complex calculations. For examples of creating widgets using scripts, see [Script Based Widgets Using Automation Scripts Examples](#).



*Cortex XSOAR supports JavaScript, Python and PowerShell.*

If you create or upload the script to the **Automation** page, you can use the script in any widget (rather than uploading the script each time), use the script with a JSON file, and you can also add it an Incident and an Indicator page.

**STEP 1 |** Create a new script by uploading or creating a new script in the **Automation** page.

You can also upload the script in the **Widgets Library**. If creating a JSON file, the script is the widget query and the script should return a value in the format of the widget type you want to use. For example, number, or text.

**STEP 2 |** For dashboards and reports, after creating the script, you need to create a widget in the **Widgets Library**, as described in [Create a Widget in the Widgets Library](#).

**STEP 3 |** Select the **Script based** data type and then add the script.

You can also create a JSON file and upload the JSON file directly to the **Widgets Library**. For information about JSON file parameters, see [JSON File Widget Parameters](#).

**STEP 4 |** Add the script based widget to one of the following:

- [Report](#)
  - [Dashboard](#)
- Ensure you select the **Script based** data type.
- [War Room](#)
  - [Incidents Page](#)
  - [Indicators Page](#)

## Script Based Widgets Using Automation Scripts Examples

The following examples are script based widgets using automation scripts. After creating the script, you do not need to create a JSON file as you can create a widget in the **Widgets Library**. For more information about creating widgets in the **Widgets Library** see [Create a Widget in the Widgets Library](#).

- [Text](#)
- [Number](#)
- [Duration](#)
- [Chart](#)
- [Table or List](#)

---

## Text

In this example, create a script that queries and returns current on-line users, and displays the data in a Markdown table. If using a JSON file, you must set `widgetType` to `text`.

In the automation script, type one of the following return values:

### JavaScript

```
return executeCommand("getUsers", {online: true})[0].HumanReadable;
```

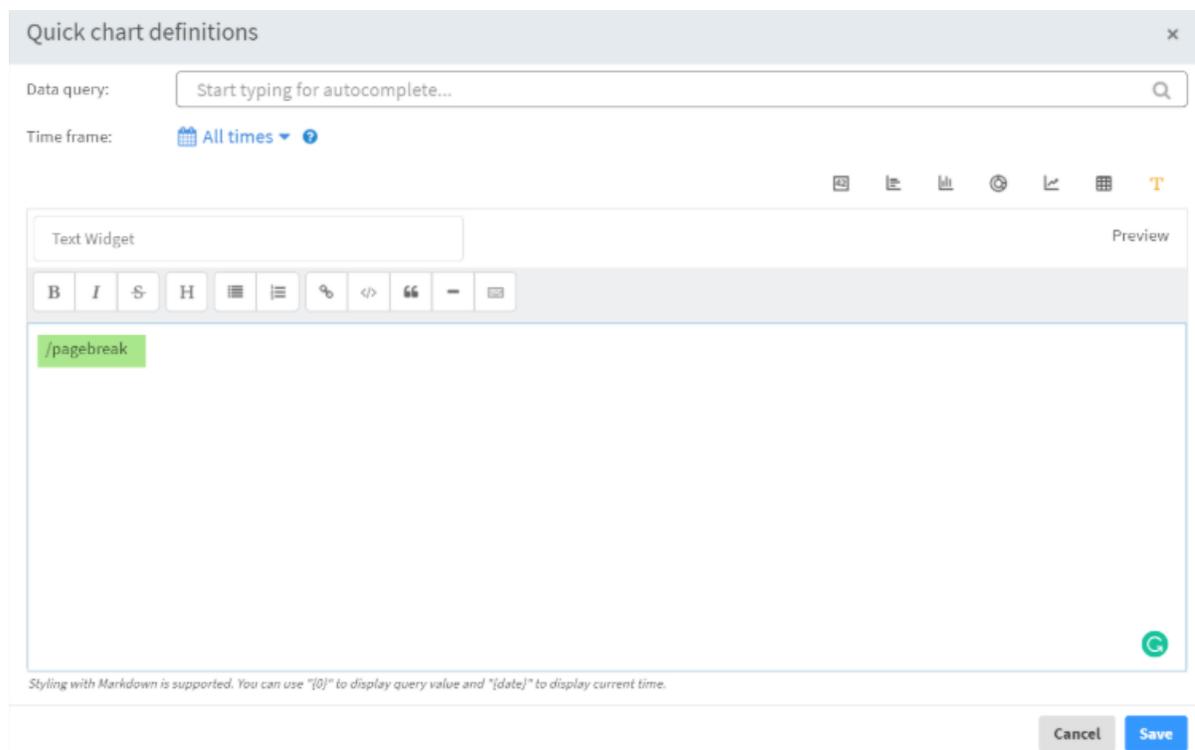
### Python

```
demisto.results(demisto.executeCommand("getUsers", { "online": True })[0]
["HumanReadable"])
```

(Optional) If creating a JSON file, type the following:

```
{
 "id": "1a2b3c4d",
 "name": "GetOnlineUsers",
 "dataType": "scripts",
 "widgetType": "text",
 "query": "GetOnlineUsers"
}
```

When creating or editing the widget in the Cortex XSOAR, to add a page break, type `/pagebreak` in the text box. When you generate a report, the widgets that follow the page break are on a separate page.



In the Cortex XSOAR dashboard, the following widget displays the on-line users:

| Online Users  |               |      |       |                               |
|---------------|---------------|------|-------|-------------------------------|
| Username      | Email         | Name | Phone | Roles                         |
| @demisto.com  | @demisto.com  |      |       | demisto: [Okta-Administartor] |
| a@demisto.com | a@demisto.com |      |       | demisto: [Okta-Administartor] |
| n@demisto.com | n@demisto.com |      |       | demisto: [Okta-Administartor] |
| s@demisto.com | s@demisto.com |      |       | demisto: [Okta-Administartor] |

## Number

This example shows how to create a single item widget with the percentage of incidents that DBot closed.

In the automation script, type one of the following:

### JavaScript

```

var res = executeCommand("getIncidents", {
 'query': 'status:closed and investigation.users:""',
 'fromdate': args.from,
 'todate': args.to,
 'size': 0
});
var closedByDbot = res[0].Contents.total;

res = executeCommand("getIncidents", {
 'status': 'closed',
 'fromdate': args.from,
 'todate': args.to,
 'size': 0
});

```

```
var overallClosed = res[0].Contents.total;

var result = Math.round(closedByDbot * 100 / overallClosed);
return isNaN(result) ? 0 : result;
```

## Python

```
res = demisto.executeCommand("getIncidents", {
 "query": "status:closed and investigation.users:\\\"\\\"",
 "fromdate": demisto.args()["from"],
 "todate": demisto.args()["to"],
 "size": 0
})
closedByDbot = res[0]["Contents"]["total"]

res = demisto.executeCommand("getIncidents", {
 "status": "closed",
 "fromdate": demisto.args()["from"],
 "todate": demisto.args()["to"],
 "size": 0
});
overallClosed = res[0]["Contents"]["total"]
if overallClosed == 0:
 demisto.results(0)
else:
 result = round(closedByDbot * 100 / overallClosed)
 demisto.results(result);
```

(Optional) If creating a JSON file, type the following:

```
{
 "id": "closed-by-dbot-incidents-percentage",
 "name": "Closed By Dbot",
 "dataType": "scripts",
 "widgetType": "number",
 "query": "DBotClosedIncidentsPercentage"
}
```

## Duration

In this example, create a script that queries and returns a time duration (specified in seconds), and displays the data as a countdown clock. If using a JSON file, you must set widgetType to duration.

In the automation script, type one of the following return values:

## JavaScript

```
return JSON.stringify([{ name: "", data: [120] }]);
```

## Python

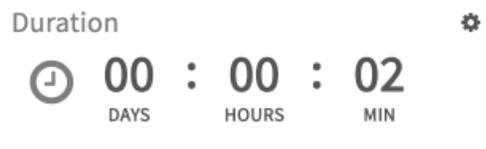
```
demisto.results(['{"name": "", "data": [120]}'])
```

The return type should be a string (any name) and an integer. The time is displayed in seconds.

(Optional) If using a JSON file, type the following:

```
{
 "id": "1a2b3c4d687",
 "name": "slaRemaining",
 "dataType": "scripts",
 "widgetType": "duration",
 "query": "RemainingSLAScript"
}
```

After you have uploaded the script and created the widget, you can add the widget to the dashboard or report. The following widget displays the time duration:



### Trend

In this example, create a script that queries and returns the trend between two sums. If creating a JSON file, set widgetType to trend.

In the automation script, type one of the following return values:

### JavaScript

```
return JSON.stringify({currSum: 48, prevSum: 32});
```

### Python

```
demisto.results({ "currSum": 48, "prevSum": 32 })
```

The returns displays a an object which compares the current sum with the previous sum.

(Optional) If creating a JSON file, type the following:

```
{
 "id": "1a2b3c4d55",
 "name": "DailyTotalTrend",
 "dataType": "scripts",
 "widgetType": "trend",
 "query": "DailyTotalTrendScript"
}
```

---

## Chart

A valid result for a chart widget is a list of groups. Each group points to a single entity, for example, in bar charts each group is a bar. A group consists of the following:

- **Name** - A string.
- **Data** - An array of integers.
- **Color** - A string representing a color that will be used as a default color for that group. It can be the name of the color, a hexadecimal representation of the color, or an rgb color value (optional).



*A widget legend color will override a group color if it exists.*

- **Groups** - A nested list of groups (optional).

In this example, we show how to create a script that will query and return the trend between two sums in a pie chart. If creating a JSON file, set the widgetType to one of the following chart types.

- Pie
- Line
- Bar
- Column

### Simple pie/chart

In the automation script, type the following return value:

#### JavaScript

```
var data = [
 {name: "2018-04-12", data: [10], color: "blue"},
 {name: "2018-04-10", data: [3], color: "#029be5"},
 {name: "2018-04-17", data: [1], color: "rgb(174, 20, 87)"},
 {name: "2018-04-16", data: [34], color: "grey"},
 {name: "2018-04-15", data: [17], color: "purple"}
];
return JSON.stringify(data);
```

#### Python

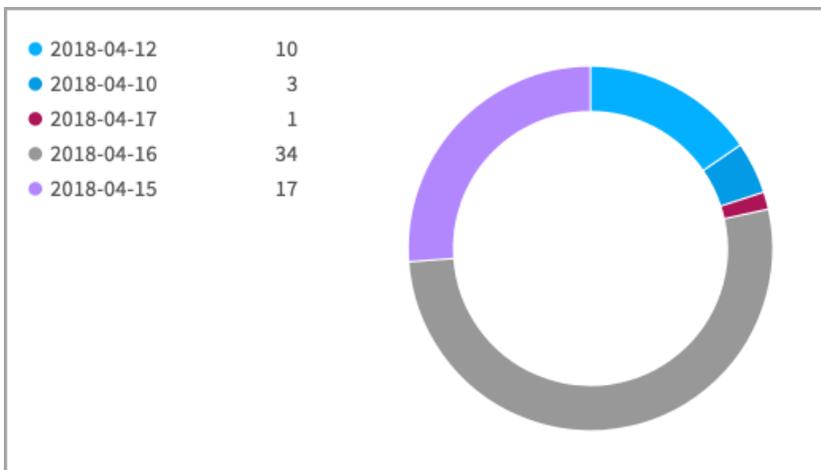
```
data = [
 {"name": "2018-04-12", "data": [10], color: "blue"},
 {"name": "2018-04-10", "data": [3], color: "#029be5"},
 {"name": "2018-04-17", "data": [1], color: "rgb(174, 20, 87)"},
 {"name": "2018-04-16", "data": [34], color: "grey"},
 {"name": "2018-04-15", "data": [17], color: "purple"}
];
demisto.results(json.dumps(data));
```

(Optional) If creating a JSON file, type the following:

```
{
 "id": "1a2b3c4dee",
 "name": "DailyTotalSales",
 "dataType": "scripts",
 "widgetType": "pie",
```

```
"query": "DailyTotalSalesScript"
}
```

After you have uploaded the script and created the widget you can add the widget to a dashboard or report. The following widget displays the trend in a pie chart:



### Two group chart

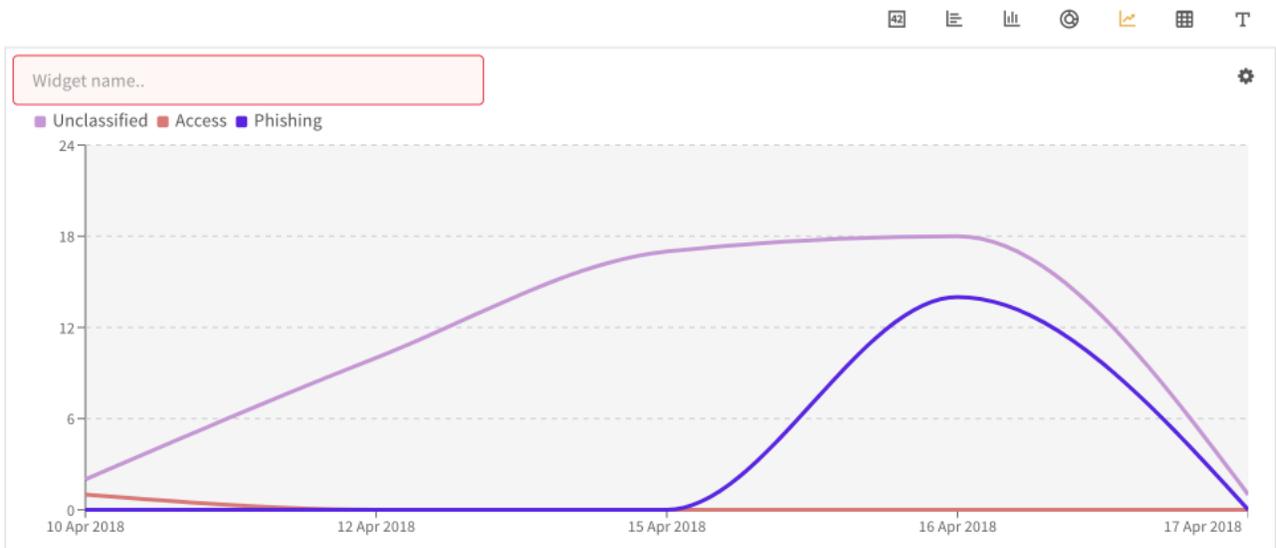
#### JavaScript

```
var data = [
 {name: "2018-04-12", data: [10], groups: [{name: "Unclassified", data:
 [10] }]}},
 {name: "2018-04-10", data: [3], groups: [{name: "Unclassified", data:
 [2] }, {name: "Access", data: [1] }]}},
 {name: "2018-04-17", data: [1], groups: [{name: "Unclassified", data:
 [1] }]}},
 {name: "2018-04-16", data: [34], groups: [{name: "Unclassified", data:
 [18] }, {name: "Phishing", data: [14] }]}},
 {name: "2018-04-15", data: [17], groups: [{name: "Access", data: [17] }]}
];
return JSON.stringify(data);
```

#### Python

```
data = [
 {"name": "2018-04-12", "data": [10], "groups": [{"name": "Unclassified",
 "data": [10] }]}},
 {"name": "2018-04-10", "data": [3], "groups": [{"name": "Unclassified",
 "data": [2] }, {"name": "Access", "data": [1] }]}},
 {"name": "2018-04-17", "data": [1], "groups": [{"name": "Unclassified",
 "data": [1] }]}},
 {"name": "2018-04-16", "data": [34], "groups": [{"name": "Unclassified",
 "data": [18] }, {"name": "Phishing", "data": [14] }]}},
 {"name": "2018-04-15", "data": [17], "groups": [{"name": "Access", "data":
 [17] }]}
];
demisto.results(json.dumps(data));
```

Time frame:  All times 



## Table or List

In this example, you need to create a script that queries and returns employee information in a table. For Table or List, if creating a JSON file, set the widgetType to table or list. When using lists, a maximum of two columns displays, the rest are ignored (do not display).

In the automation script, type one of the following return values:

### JavaScript

```
return JSON.stringify({total: 3, data:[
 {Employee: 'David D', Phone: '+14081234567', Email: 'David@org.com'},
 {Employee: 'James J', Phone: '+14087654321', Email: 'James@org.com'},
 {Employee: 'Alex A', Phone: '+14087777777', Email: 'Alex@org.com'}
]});
```

### Python

```
demisto.results({"total": 3, "data": [{"Employee": "David D", "Phone":
"+14081234567", "Email": "David@org.com"}, {"Employee": "James J", "Phone":
"+14087654321", "Email": "James@org.com"}, {"Employee": "Alex A", "Phone":
"+14087777777", "Email": "Alex@org.com"}]})
```

(Optional) If creating a JSON file, type the following:

```
{
 "id": "1a2b3c4de345",
 "name": "EmployeeInfo",
 "dataType": "scripts",
```

```
"widgetType": "table",
"query": "EmployeeInfoScript"
}
```

After you have uploaded the script and created a widget you can add the widget to a dashboard or report. The following widget displays the employee information:

| Employee | Email         | Phone        |
|----------|---------------|--------------|
| David D  | David@org.com | +14081234567 |
| James J  | James@org.com | +14087654321 |
| Alex A   | Alex@org.com  | +14087777777 |

## Create a Widget from an Indicator

Widgets are the visual components that populate dashboards and reports with specific data. Although there are various out-of-the-box system widgets available, you can create custom widgets from indicators and then add them to a dashboard or report. You can also [Create a Widget in the Widgets Library](#).

To create a widget from an indicator, you need to run a query from the Indicators page, and then save the visual results as a widget.

**STEP 1** | In the **Indicators** page, from the drop down list select the date range.

**STEP 2** | In the query field, type the query criteria as required and run the query.

**STEP 3** | Click **Create a Widget** button.

**STEP 4** | Add the [Widget Parameters](#).

**STEP 5** | Click **Save**.

The widget is added to the **Widgets Library**.



*By default, the widget inherits the date range that you specify when creating the widget, but you can modify the date range when you create the dashboard or report. If the date range for the report or dashboard does not include the widget date range, the data is blank. To override the dashboard or report's date range, click Use Widget's date range.*

## Add a Custom Widget to the Indicator Page

You can add a custom widget to the **info** tab when you view a detailed indicator. Every time you open the detailed incident or indicator type you can see the widget.

Before you start, you need to create a custom widget, as described in [Create a Custom Widget Using an Automation Script](#).



*In the Automation page, when adding or editing the script you want to use, ensure that you add the dynamic-section label.*

**STEP 1** | Go to **Settings > Advanced**.

**STEP 2** | To add a widget to an indicator, click the **Indicator types** tab.

---

**STEP 3** | Select the indicator type you want the widget to appear and click **Edit layout**.

**STEP 4** | From the **Layout Builder** window, in the **Library** section, drag the **General Purpose Dynamic Section** into the layout area you want it to appear.

**STEP 5** | In the **General Purpose Dynamic Section**, click the edit button.

**STEP 6** | Type a name and description for the widget.

**STEP 7** | In the **Automation script** field, from the drop down list select the automation script you want to add.



*If the automation script does not appear, you need to add the dynamic-section label to the script in the Automation page.*

**STEP 8** | Click **OK**.

For an example of adding a widget to a incident (which follows the same procedure), see [Add a Custom Widget to the Incident page](#).

---

# Edit a Widget

You can edit an existing widget in the dashboard or report, or in the **Widgets Library**. If editing a widget in the **Widgets Library** it is available to all users. If editing a widget in a dashboard or a report directly the original widget in the **Widgets Library** is unaffected.

Not all widgets are editable, such as system widgets.

**STEP 1** | Create or edit a dashboard to report.

**STEP 2** | To edit the widget in a dashboard to report, from the widget, select the **gear icon** > **Edit Widget**.

If the widget is not in a dashboard or report, you need to add the widget.

**STEP 3** | To edit the widget in the **Widgets Library**, search for the widget and then click the edit button.

**STEP 4** | In the **Quick chart definitions** window, edit the [Widgets Parameters](#) as required.

**STEP 5** | Click **Save**.

If you are editing the widget in the dashboard or report, the widget appears in the dashboard or report. You can then adjust the size and move the widget as required.

---

# Create a Used Percentage Widget for a Disk Partition

In the **System Health** tab, by default you can view the used percentage widget for the whole disk. If you want to create a usage percentage widget for a disk partition, you need to add a server configuration and then create a new dashboard.

**STEP 1** | Add the partition path to the Server configuration.

1. Select **Settings > About > Add Server Configuration**.
2. Add the following key and value:

| Key                                     | Value                                                                                                             |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <code>disk.partitions.to.monitor</code> | <i>Path and name of the partition.</i> For example, if your disk partition is <code>/run</code> , add this value. |

Wait a few minutes for the system to update.

**STEP 2** | Export the **System Health** dashboard.

1. Go to **Home > DASHBOARDS > System Health**.
2. Click  and then click **Export**.

The dashboard file downloads.

**STEP 3** | Update the dashboard file you downloaded in step 2.2.

1. Open the file, search for `"query": "disk.usedPercent"`, and add the following disk partition:  
`"query": "disk.usedPercent./<name of partition>"`,  
For example, to add, `/run`, change the parameter to `"query": "disk.usedPercent./run"`,

```
"fromDateLicense": "0001-01-01T00:00:00Z",
"name": "System Health",
"prevName": "System Health",
"layout": [
 {
 "id": "2a235a60-2ead-11e8-be34-cd9a232a8686",
 "forceRange": false,
 "x": 0,
 "y": 0,
 "i": "2a235a60-2ead-11e8-be34-cd9a232a8686",
 "w": 2,
 "h": 3,
 "widget": [
 {
 "id": "3",
 "version": 1,
 "modified": "2018-03-23T15:16:10.634227Z",
 "sortValues": null,
 "vcShouldIgnore": false,
 "commitMessage": "",
 "shouldCommit": false,
 "name": "Disk Used Percentage",
 "prevName": "",
 "dataType": "system",
 "widgetType": "number",
 "query": "disk.usedPercent./run",
 "sort": null,
 "isPredefined": true,
```

2. Save the file as a yml file.

#### STEP 4 | Import the file.

1. In the **DASHBOARDS** tab, click  and then click **Import**.
2. Select the file and click **Open**.

The **Disk Used Percentage** widget for the disk partition appears in the new dashboard. It may take a short while for the widget to update.

# Saved By Dbot (ROI) Widget

In the Dashboard, **Incidents** tab, Cortex XSOAR comes with a number of pre-installed widgets, such as **Saved by DBot**.

The **Saved by Dbot** widget calculates the amount saved in Dollars according to actions carried out by all users in Cortex XSOAR across all incidents.



*Although the widget comes out of the box with Cortex XSOAR, you can add the Return on Investment (ROI) widget in the Widgets Library, which is identical to the Saved by Dbot widget.*

The following parameters are used to calculate the amount saved by Dbot (ROI):

| Parameter | Description                                                                                                                                                |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Man Hour  | The amount in Dollars of each hour for an analyst.                                                                                                         |
| Roundtrip | The time it takes in minutes to run an integration task with any of the integrated products. This can be a command within a script or inside the War Room. |
| Report    | The times it takes to write an incident report.                                                                                                            |
| Script    | The time it takes to undertake an action that a script would do.                                                                                           |

The ROI is calculated as follows:

```
#of times roundtrip done in the given time period * time taken to do roundtrip) + (# of times report generated in the given time period * time taken to generate report) + (# of times automation run in given time period * time taken to run automation)] * cost of 1 Man hour in dollars.
```

You can change the way ROI is calculated based on your own statistics of time taken to perform the tasks for the actions when done manually. To change the statistics, select **Settings > About > Troubleshooting > Add Server Configuration** and add the following keys and values:

| Keys                            | Values                                       |
|---------------------------------|----------------------------------------------|
| <code>ROI.Cost.ManHour</code>   | Amount in Dollars. Default: 60               |
| <code>ROI.Time.RoundTrip</code> | Amount in minutes (whole number). Default: 5 |
| <code>ROI.Time.Report</code>    | Amount in minutes (whole number). Default: 5 |
| <code>ROI.Time.Script</code>    | Amount in minutes (whole number). Default: 5 |

You can also change the [currency symbol](#) from Dollars to a currency of your choice.

---

## Customize the Currency Symbol in the Saved by Dbot Widget

The default currency symbol in the Saved by Dbot widget is the Dollar sign (\$). To change the currency symbol, you need to [create a widget using a JSON file](#).

In this example, which you can use as a template, we changed the value for the `currencySign` argument to Euro (€).

```
{
 "size":5,
 "dataType":"roi",
 "params":{
 "currencySign":"€"
 },
 "query":"","
 "modified":"2019-01-12T15:13:09.872797+02:00",
 "shouldCommit":false,
 "name":"Return On Investment (ROI)",
 "shouldPush":false,
 "dateRange":{
 "fromDate":"0001-01-01T00:00:00Z",
 "toDate":"0001-01-01T00:00:00Z",
 "period":{
 "by":"","
 "byTo":"","
 "byFrom":"days",
 "toValue":null,
 "fromValue":30,
 "field":""
 }
 },
 "commitMessage":"","
 "isPredefined":true,
 "version":13,
 "id":"roi",
 "shouldPublish":false,
 "category":"others",
 "sort":null,
 "prevName":"Return On Investment (ROI)",
 "widgetType":"number"
}
```

# Manage Indicators

- > Understand Indicators
- > Indicators Page
- > Indicator Reputation
- > Indicator Types
- > Indicator Fields
- > Exclusion List
- > Create a Feed-Triggered Job
- > Manage the Indicator Timeline
- > Auto Extract Indicators



---

# Understand Indicators

Indicators are artifacts associated with incidents, and are an essential part of the incident management and remediation process.

They help to correlate incidents, create hunting operations, and enable you to easily analyze incidents and reduce MTTR.

Cortex XSOAR includes an Indicator repository, which collects and correlates indicators across all incidents, alerts, and feeds flowing into Cortex XSOAR.

- [Indicators Page](#)
- [Indicator Reputation](#)
- [Indicator Types](#)
- [Indicator Fields](#)

## Detect and ingest indicators

There are several methods by which indicators are detected and ingested in Cortex XSOAR.

| Method      | Description                                                                                                                                                                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Integration | <ul style="list-style-type: none"><li>• Feed: integrations that <a href="#">fetch indicators from a feed</a>, for example TAXII, AutoFocus, Office 365, and so on.</li><li>• Mail: integrations that consume emails with STIX or CSV files and add the indicators to the indicator repository.</li></ul> |
| Incident    | <ul style="list-style-type: none"><li>• Manual: user marks a piece of data as an indicator.</li><li>• Auto-extract: indicators are extracted from every incident that flows into Cortex XSOAR, for example from a SIEM integration.</li></ul>                                                            |
| Regex query | A query that identifies indicators in the War Room.                                                                                                                                                                                                                                                      |
| STIX file   | Manually upload a STIX file on the Indicators page.                                                                                                                                                                                                                                                      |
| Script      | <ul style="list-style-type: none"><li>• <b>FetchIndicatorsFromFile</b>: accepts a file from which it extracts indicators and processes them in Cortex XSOAR.</li><li>• <b>CreateIndicatorsFromSTIX</b>: extracts indicators from a STIX file and processes them in Cortex XSOAR.</li></ul>               |

## Feed Integrations

Cortex XSOAR has several out-of-the-box threat intelligence feed integrations.

- AutoFocus
- AWS
- Microsoft Azure
- Bambenek Consulting
- Blocklist\_de
- Microsoft Office 365
- Palo Alto Networks PAN-OS EDL Service
- Proofpoint
- Recorded Future RiskList

- Spamhaus
- TAXII

### Common feed integration parameters

This is a non-exhaustive list of the most common feed integration parameters. Each feed integration might have parameters unique to that integration. Make sure to read the documentation for specific feed integrations.

| Parameter                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                        | A meaningful name for the integration instance. For example, if you have separate instances to fetch indicator types, you can include the name of the indicator type that the instance fetches.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Fetch indicators            | Select this option for the integration instance to fetch indicators.<br><br>Some integrations can fetch indicators or incidents. Make sure you select the relevant option for what you need to fetch in the instance.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Sub-Feeds                   | Some feeds might have several lists or files that provide indicators. The sub-feeds parameter enables you to select the specific list or file from which to fetch indicators. For example, Bambenek Consulting provides different lists for IPs and domains. Each of the Bambenek lists are available as sub-feeds.                                                                                                                                                                                                                                                                                                                                                                    |
| URL                         | The URL of the feed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Fetch Interval              | How often the integration instance should fetch indicators from the feed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Indicator Reputation        | The <a href="#">indicator reputation</a> to apply to all indicators fetched from this integration instance.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Source Reliability          | The <a href="#">reliability of the source</a> providing the threat intelligence data.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Indicator Expiration Method | The method by which to expire indicators from this integration instance. The default expiration method is the interval configured for the indicator type to which this indicator belongs. <ul style="list-style-type: none"> <li>• Indicator Type: the expiration method defined for the indicator type to which this indicator belongs (interval or never).</li> <li>• Time Interval: expires indicators from this instance after the specified time interval, in days or hours.</li> <li>• Never Expire: indicators from this instance never expire.</li> <li>• When removed from the feed: when the indicators are removed from the feed they are expired in the system.</li> </ul> |
| Bypass exclusion list       | When selected, the exclusion list is ignored for indicators from this feed. This means that if an indicator from this feed is on the exclusion list, the indicator might still be added to the system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Trust any certificate       | When selected, certificates are not checked.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| Parameter                 | Description                                                                                                        |
|---------------------------|--------------------------------------------------------------------------------------------------------------------|
| Use system proxy settings | Runs the integration instance using the proxy server (HTTP or HTTPS) that you defined in the server configuration. |
| Do not use by default     | Excludes this integration instance when running a generic command that uses all available integrations.            |

## Indicators Page

The Indicators page displays indicator dashboards, a table or summary view of all indicators, and enables you to perform several indicator actions.

### Indicator actions

You can perform the following actions on the Indicators page.

| Action                 | Description                                                                                                                                                                                                                                 |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create incident        | Creates an incident from the selected indicators and populates relevant incident fields with indicator data.                                                                                                                                |
| Edit                   | You can edit a single indicator or select multiple indicators to perform a bulk edit.                                                                                                                                                       |
| Delete and Exclude     | You can select to delete and exclude one or more indicators from all indicator types or from a subset of indicator types.<br><br>If you select the <b>Do not add to exclusion list</b> check box, the selected indicators are only deleted. |
| Export                 | Exports the selected indicators to a CSV file.                                                                                                                                                                                              |
| Export (STIX)          | Exports the selected indicators to a STIX file.                                                                                                                                                                                             |
| Upload a STIX file     | Uploads a STIX file and adds the indicators from the file to the system.                                                                                                                                                                    |
| Create a new indicator | Manually creates a new indicator in the system.                                                                                                                                                                                             |

### Indicator query

You can search for indicators using any of the available search fields, but there are several fields specific to indicators that you can use to search for indicators.

| Field            | Description                                                         |
|------------------|---------------------------------------------------------------------|
| sourceBrands     | Indicator feed or enrichment integrations.                          |
| sourceInstances  | A specific instance of an indicator feed or enrichment integration. |
| expirationSource | The source of the indicator having expired status.                  |

| Field    | Description                                                             |
|----------|-------------------------------------------------------------------------|
| isShared | Whether the indicator is shared to tenant accounts (multi-tenant only). |
| tags     | Tags applied to indicators.                                             |
| comments | Search for keywords within indicators' comments.                        |

## Indicator Reputation

An indicator's reputation is assigned according to the reputation returned by the source with the highest reliability. In cases where multiple sources with the same reliability score return a different reputation for the indicator, the worst reputation is taken.

### Indicator reputations

Indicators are assigned a reputation on a scale of 0 to 3.

| Score | Reputation | Color    |
|-------|------------|----------|
| 0     | None       | No color |
| 1     | Good       | Green    |
| 2     | Suspicious | Orange   |
| 3     | Bad        | Red      |

### Example 1

In this example, two 3rd-party integrations, VirusTotal and AlienVault, return a different reputation for the same indicator. VirusTotal returns a reputation of Good, and AlienVault returns a reputation of Bad. The indicator's reputation will be Bad.

### Example 2

In this example, two sources with different reliability scores return a different reputation for the same indicator. The first source is a TAXII feed with a reliability score of C - Fairly reliable, and the second source is a CSV feed with a reliability score of B - Usually reliable. The TAXII feed returns a reputation of Bad and the CSV feed returns a reputation of Good. The indicator's reputation will be Good because the CSV reliability score is higher than that of the TAXII feed.

### Source reliability

The reliability of an intelligence-data source influences the reputation of an indicator and the values for indicator fields when merging indicators.

Indicator fields are merged according to the source reliability hierarchy. This means that when there are two different values for a single indicator field, the field will be populated with the value provided by the source with the highest reliability score.

In rare cases, two sources with the same reliability score might return different values for the same indicator field. In these cases, the field will be populated with the most recently provided source.

For the field types Tags and Multi-select, all values are appended, nothing is overridden.

| Source               | Reliability Score               | Notes                                                                                                                                                                                      |
|----------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| User (manual)        | A+++                            | A user manually updates the reputation of an indicator.                                                                                                                                    |
| Reputation script    | A++                             | A script with the <b>reputation</b> tag, which calculates the reputation of an indicator. For example, the <i>DataDomainReputation</i> script evaluates the reputation of a URL or domain. |
| 3rd-party enrichment | A+                              | An integration or service that evaluates the reputation of an indicator. For example, the <i>urlscan.io</i> integration evaluates the reputation of a URL.                                 |
| Feed reliability     | A: Completely reliable          | The feed reliability is applied at the integration instance level.                                                                                                                         |
|                      | B: Usually reliable             |                                                                                                                                                                                            |
|                      | C: Fairly reliable              |                                                                                                                                                                                            |
|                      | D: Not usually reliable         |                                                                                                                                                                                            |
|                      | E: Unreliable                   |                                                                                                                                                                                            |
|                      | F: Reliability cannot be judged |                                                                                                                                                                                            |

### Indicator expiration

Indicators can have the status Active or Expired, which is determined by the **expirationStatus** field. When indicators expire, they still exist in Cortex XSOAR, meaning they are still displayed and you can still search for them. A job runs every hour to check for newly expired indicators.

By default, indicators are expired according to either the expiration interval configured for the indicator type to which the indicator belongs, or to never expire.

This is the hierarchy by which indicators are expired.

| Method           | Description                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Manual           | A user manually expires an indicator. This method overrides all other methods.                                                                                    |
| Feed integration | The expiration method configured for an <a href="#">integration instance</a> , which overrides the method defined for the indicator type.                         |
| Indicator type   | The expiration method defined for the indicator type to which this indicator belongs (interval or never). This is the default expiration method for an indicator. |

---

## Customize the Dbot Reputation Score Logic

Cortex XSOAR calculates the reputation score of an entity (IP, URL or hash file) by considering the score received by third-party integrations used (such as VirusTotal and X-Force). The Dbot score is determined by the data reputation scripts. Amending the following scripts changes the way Dbot handles the reputation.

- **DataIPReputation**
- **DataHashReputation**
- **DataURLReputation**

You can customize the Dbot reputation score according to your own logic. For information about how Cortex XSOAR handles reputation scores, see [Indicator Reputation](#).

**STEP 1** | Go to the **Automation** page and locate the script you want to edit.

**STEP 2** | Click **Copy Automation** and modify an existing reputation script, such as **DataURLReputation**.

In the following example, we redefine the values for each reputation:

```
var score = -1;
var rep = executeCommand('ip', {ip: args.input});
var scores = dq(rep, 'EntryContext.dbotScore.score');
for (var i = 0; scores && i < scores.length; i++) {
 if (scores[i] && score < scores[i]) {
 score = scores[i];
 }
}
if (score != 1) {
 score = 3;
}
return score < 0 ? 0 : score;
```

Ensure that the **Reputation** tag is selected.

**STEP 3** | Click **Save**.

**STEP 4** | To add the script to the indicator, go to **Settings > Indicator Types**.

**STEP 5** | Select the indicator type that you want to add the script and click **Edit**.

**STEP 6** | In the **Reputation Script** field, select the script you modified in step 2.

**STEP 7** | Click **Save**.

## Indicator Types

The indicators are categorized by indicator type, which determines the indicator layout (fields) that are displayed and which scripts are run on indicators of that type.

There are several system-level indicator types.

- IP Address
- Registry Path Reputation
- File
- Email
- Username
- Hostname
- Domain
- File Enhancement Scripts

- CVE CVSS Score

## Create an Indicator Type

When you create a custom indicator type, you configure numerous fields and settings that impact how indicators of that type are enriched, expired, how the reputation is calculated, among others.

Before you create a custom indicator type, you should familiarize yourself with the [indicator type profile](#).

**STEP 1** | Go to **Settings > Advanced > Indicator Type**.

**STEP 2** | Click the **Add Indicator Type** button.

**STEP 3** | Configure the indicator type as needed.

## Indicator Type Profile

In addition to configuring the standard indicator type fields, you can [map custom indicator fields](#) to context data.

There are a number of configuration options and fields that you must complete when creating a new indicator type.

**Table 1: Settings**

| Field                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                  | A meaningful name for the indicator type.                                                                                                                                                                                                                                                                                                                                                                                          |
| Regex                 | The regular expression (regex) by which to identify indicators for this indicator type.                                                                                                                                                                                                                                                                                                                                            |
| Formatting Script     | The script to run on and modify how the indicator displays in Cortex XSOAR, such as in the War Room, reports, and so on. For example, the UnescapeURLs script extracts URLs that are redirected by security tools or unescapes URLs that are escaped for safety (e.g., <code>hxxps://www[.]CortexXSOAR[.]com</code> ).                                                                                                             |
| Enhancement Scripts   | A script to run on an identified indicator. For example, an enrichment script, a script that runs a search in a SIEM for the indicator, and so on.<br><br>After indicators are identified, you can go to the indicator quick view, click the <b>Actions</b> button and run an enhancement script directly on an indicator. In order for these scripts to be available in the drop-down menu, they need the <b>enhancement</b> tag. |
| Reputation Command    | The command to run to calculate the reputation of indicators of this type. The result (reputation) is only associated with the specific indicator on which it's run (not the indicator type).                                                                                                                                                                                                                                      |
| Excluded Integrations | Integrations to exclude when calculating the reputation, evaluating, and enriching indicators of this indicator type.                                                                                                                                                                                                                                                                                                              |
| Reputation Script     | User-created scripts that either override the Cortex XSOAR command algorithm or run on top of the data returned from the command. In order for these scripts to be available in the drop-down menu, they                                                                                                                                                                                                                           |

| Field                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                       | require the <b>reputation</b> tag. The output of this script is a reputation score, which is used as the basis for the indicator reputation.                                                                                                                                                                                                                                                                                                                                                                                           |
| Indicator Expiration Method                           | <p>The method by which to expire indicators of this type. The expiration method that you select is the default expiration method for indicators of this indicator type.</p> <p>The expiration can also be assigned when configuring a <a href="#">feed integration instance</a>, which overrides the default method.</p> <ul style="list-style-type: none"> <li>• Never Expire: indicators of this type never expire.</li> <li>• Time Interval: indicators of this type expire after the specified number of days or hours.</li> </ul> |
| Context path for reputation value ( <i>Advanced</i> ) | When an indicator is auto-extracted, the entry data from the command is mapped to the incident context. This path defines the context key that the indicator reputation is mapped to.                                                                                                                                                                                                                                                                                                                                                  |
| Context value of reputation ( <i>Advanced</i> )       | The value of this field defines the actual data that is mapped to the context path.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Cache expiration in minutes ( <i>Advanced</i> )       | The amount of time (in minutes) after which the cache for indicators of this type expire. The default is 4,320 minutes (three days).                                                                                                                                                                                                                                                                                                                                                                                                   |



*Formatting scripts for out-of-the-box indicator types are now system level. This means that the formatting scripts for these indicator types are not configurable. To create a formatting script for an out-of-the-box indicator type, you need to disable the existing indicator type and create a new (custom) indicator type. If you configured a formatting script before this change and updated your content, this configuration will revert to content settings (empty).*

## File Indicators

Cortex XSOAR uses a single File indicator for file objects. Files appear with their SHA256 hash and all other hashes associated with the file, (MD5, SHA1, and SSDeep) are listed as properties of the same indicator. Also, when ingesting an incident through an integration, all file information is presented as one object.

For example, when viewing an incident, you can see a file indicator with a **Bad** Reputation value:

| Type | Value                                                               | Name | Reputation ↓ | First Seen             | Last Seen                       |
|------|---------------------------------------------------------------------|------|--------------|------------------------|---------------------------------|
| File | 787bab5ae486998a8c4efe1cbfcd5b252023a00<br>ffd78fdc9876aa6cb17779ed |      | Bad          | August 1, 2019 5:43 PM | 181@1<br>August 1, 2019 5:52 PM |

When clicking the indicator, you can see additional information for that indicator, including all of the other known hashes associated with this file:

Indicator Quick View
✕

●  
**File**

✎
Actions

SHA256: 787bab5ae486998a8c4efe1cbfcdd5b252023a00ffd78fd...

**First Seen:**  
August 1, 2019 5:43 PM

**Last Seen:**  
August 1, 2019 5:52 PM

---

Bad Set by @DBot

**Hashes**

MD5  
9ACE54381A3072921A37142B6A3A5191

SHA1  
4FC56741E3794A8C977239B88BA7E647ECE24E0A

SHA256  
787bab5ae486998a8c4efe1cbfcdd5b252023a00ffd78fd9876aa6cb17779ed

If the file appears in a different incident with a different name, and has any of the same hash values, it automatically associates with the original indicator.

 *The new File indicator only affects new indicators ingested to the Cortex XSOAR platform. Indicators that were already in Cortex XSOAR continue to appear as their respective hash-related indicators.*

If you want to have each file hash appear as its own indicator, do the following:

1. Go to **Settings > Advanced > Indicator Types**.
2. Select the **File** indicator and click **Disable**.
3. Select the following required hashes:
  - File SHA-256
  - File SHA-1
  - File MD5
  - SSDeep
4. Click **Enable**.

## Indicator Fields

After you create a custom indicator field, you can add it to the indicator layout for the indicator types to which you associated the field.

- [Create a Custom Indicator Field](#)
- [Map Custom Indicator Fields](#)

### Create a Custom Indicator Field

Indicator Fields are used to add specific indicator information to incidents. When you create an indicator field, you can associate the field to a specific incident type or to all incident types.

**STEP 1 |** Go to **Settings > Advanced > Fields**.

**STEP 2 |** From the drop-down menu, select **Indicator**.

---

**STEP 3 | Click New Field.**

**STEP 4 | Configure the basic settings.**

| Field          | Description                                                                                                                                                                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Field Type     | Determines the acceptable values for the field.                                                                                                                                                                         |
| Case Sensitive | If selected, the field is case sensitive, which affects searching for the field in Cortex XSOAR.                                                                                                                        |
| Mandatory      | If selected, this field is mandatory when used in a form.                                                                                                                                                               |
| Field Name     | A meaningful display name for the field. After you type a name, you will see below the field that the <b>Machine name</b> is automatically populated. The field's machine name is applicable for searching and the CLI. |
| Tooltip        | An optional tooltip for the field.                                                                                                                                                                                      |
| Placeholder    | Optional text to display in the field when it is empty.                                                                                                                                                                 |

**STEP 5 | Configure the attributes.**

| Field                          | Description                                                                                                                                                                                                           |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add to indicator types         | By default, the <b>Associate to all</b> option is selected, which means this field will be available to use in all incident types.<br><br>Clear the check box to associate this field to a subset of indicator types. |
| Make data available for search | The values for this field can be returned in searches.                                                                                                                                                                |

**STEP 6 | Click Save.**

## *Map Custom Indicator Fields*

You map custom fields for an indicator type. Before you can map custom indicator fields, you need to [Create a Custom Indicator Field](#) and associate the field with the necessary indicator types.

**STEP 1 | Go to Settings > Advanced > Indicator Types.**

**STEP 2 | Select the check box for the indicator for which to map the custom fields.**

**STEP 3 | Click the Edit button.**

**STEP 4 | Click the Custom Fields tab.**

The custom fields associated with this incident type are listed in the table. If you do not see a custom field in the list, verify that you associated the custom field to this incident type.

**STEP 5 | (Optional) In the Indicator Sample panel, enter an indicator relevant to the indicator type to load sample data.**

---

You can map a field to a context key from the indicator's returned context data.

**STEP 6 |** Click **Choose data path** to map the custom field to a data path.

1. (Optional) Click the curly brackets to map the field to a context path.
2. (Optional) From the **Indicator Sample** panel, select a context key to map to the field.

## Exclusion List

Indicators added to the exclusion list are ignored by the system and are not considered indicators. You can still manually enrich IP addresses and URLs that are on the exclusion list, but the results are not posted to the War Room.

There are several methods by which to add indicators to the exclusion list.

### *Delete and exclude*

You can select one or more indicator from the Indicators table and click the **Delete and Exclude** button. The indicators are deleted from the Indicators table and added to the exclusion list. You can associate these indicators with one or more indicator types.

### *Manually add indicators to the exclusion list*

From the **Exclusion List** page, you can manually add a single indicator or define indicators using a regular expression (regex) or CIDR.

#### **Regex**

A regular expression enables you to identify a sequence of characters in an unknown string. The following example would identify `www.demisto.com`: `[A-Za-z0-9!@#%$.&]*demisto[A-Za-z0-9!@#%$.&]*`.

#### **CIDR**

Classless inter-domain routing (CIDR) enables you to define a range of IP addresses. For example, `192.168.100.14/24` represents the IPv4 address `192.168.100.14` and its associated routing prefix `192.168.100.0`, or equivalently, its subnet mask `255.255.255.0`, which has 24 leading 1-bits. The IPv4 block `192.168.100.0/22` represents the 1024 IPv4 addresses from `192.168.100.0` to `192.168.103.255`.

## Create a Feed-Triggered Job

You can define a job to trigger a playbook when the specified feed or feeds finish a fetch operation that included a modification to the feed. The modification can be a new indicator, a modified indicator, or a removed indicator.

For example, you want to update your firewall every time a URL is added to, modified, or removed from the Office 365 feed. You can configure a job that triggers that playbook to run whenever a modification is made to that feed.

You can customize the new job form by editing the *Indicator Feed* incident type.



*If you want to trigger a job after a feed completes a fetch operation, and the feed does not change frequently, you can select the **Reset last seen** option in the feed integration instance. The next time the feed fetches indicators, it will process them as new indicators in the system.*

**STEP 1 |** Go to the **Jobs** section.

**STEP 2** | Click the **New Job** button.

**STEP 3** | Configure the job parameters.

| Parameter | Description                                                                                                                                                                                                                                                                                                                           |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Job type  | Select the <b>Feed triggered</b> option.                                                                                                                                                                                                                                                                                              |
| Trigger   | Define the trigger for the playbook. <ul style="list-style-type: none"><li>• <b>All feeds:</b> the playbook will run when a modification is made to any feed.</li><li>• <b>Specific feeds:</b> select the feed instances that will trigger the playbook to run when a modification is made to the specified feed instances.</li></ul> |
| Name      | A meaningful name for the job.                                                                                                                                                                                                                                                                                                        |
| Playbook  | The playbook that will run when the conditions for the job are met.                                                                                                                                                                                                                                                                   |
| Tags      | Add tags to apply to the job, which you can use as a search parameter in the system.                                                                                                                                                                                                                                                  |

## Manage the Indicator Timeline

A large number of indicators can affect performance of the indicator timeline. There are several advanced server configurations you can implement to manage the indicator timeline performance.

**STEP 1** | Go to **Settings > About > Troubleshooting**.

**STEP 2** | In the **Server Configuration** section, click **Add Server Configuration**.

| Key                                                                                             | Value                                   | Description                                                                                                                                                                                                                  |
|-------------------------------------------------------------------------------------------------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>indicator.timeline.enabled</code>                                                         | <code>true</code> or <code>false</code> | Enables the indicator timeline in all flows. The default is <code>true</code> .                                                                                                                                              |
| <code>indicator.timeline.enabled.type</code> or <code>indicator.timeline.enabled.type.ip</code> | <code>true</code> or <code>false</code> | Enables the indicator timeline for a specific indicator type. This configuration overrides the <code>indicator.timeline.enabled</code> configuration.<br><br>For example:<br><code>indicator.timeline.enabled.type.ip</code> |
| <code>indicator.timeline.auto.extract.enabled</code>                                            | <code>true</code> or <code>false</code> | Enables the indicator timeline in the auto-extract flow. The default is <code>true</code> .                                                                                                                                  |
| <code>indicator.timeline.max.size</code>                                                        | Number                                  | The maximum number of indicator comments (timeline and regular). The default is 100.                                                                                                                                         |
| <code>indicator.timeline.worker.enabled</code>                                                  | <code>true</code> or <code>false</code> | Enables you to add timeline comments through content integrations.                                                                                                                                                           |

# Auto Extract Indicators

The Auto Extract feature extracts indicators and enriches their reputations using commands and scripts defined for the indicator type. You can automatically extract indicators in the following scenarios:

- Incident creation
- In a playbook task
- Using the command line

By default, Auto Extract is enabled for incident creation and using the command line to help you get up and running as you set up your environment. As your system matures and you start ingesting more events and have more integrations configured, using Auto Extract can adversely affect system performance.

As a result, Cortex XSOAR recommends that you turn off Auto Extract using the server configurations for the different Auto Extract options and only turn it on for those specific scenarios where it is necessary.

## Auto Extract Modes

Auto Extract supports the following modes:

- None - Indicators are not automatically extracted. Use this option when you do not want to further evaluate the indicators.
- Inline - Indicators are extracted and enriched within the context that Auto Extract runs, and the findings are added to the Context Data. For example, if you define Auto Extract for the Phishing incident type as inline, all of the indicators for incident classified as Phishing will be extracted and enriched before anything else happens. The playbook you defined to run by default will not run until the indicators have been fully processed. Use this option when you need to have the most robust information available per indicator. Unless otherwise configured in a system configuration, this is the default mode in which Auto Extract executes.



*This configuration may delay playbook execution (incident creation).*

- Out of band - Indicators are enriched in parallel (or asynchronously) to other actions. The enriched data is available within the incident, however, it is not available for immediate use in task inputs or outputs since the information is not available in real time.

## Global Server Configurations for Auto Extract

You can control the default behavior for auto extract using the following server configurations:

| Component         | Key                                                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Incident creation | <b>reputation.calc.algorithm:</b> applies to incident creation generally. Default is inline. You can change the value when editing an incident type, which overrides the system configuration for this incident type. |
| Tasks             | <b>reputation.calc.alogorithm.tasks:</b> applies to the result of the task. Default is none. You can change the value when editing a task, which overrides the system configuration for this task.                    |
| Manual            | <b>reputation.calc.algorithm.manual:</b> applies to commands triggered from the CLI. Default is Out of Band.                                                                                                          |

| Component | Key                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------|
|           | You can change the value when using the <b>auto-extract</b> parameter, which overrides the system configuration for this parameter. |

Each configuration can accept one of the following values:

- 1 = None
- 2 = Inline.
- 3 = Out of Band

## How to Define Auto Extract

### Incident Types

To define auto extract for an incident type, do the following:

1. Navigate to **Settings > Advanced > Incident Types**.
2. Select the incident you want to edit by clicking the checkbox and then clicking the **Edit** button.
3. In the auto extract drop down menu, select the mode you want to use.

If you select **Use system default**, you use the values defined in the [server configurations](#).

4. Click **Save**.

For an example on how to use Auto Extract, see [Auto Extract Indicators from a Phishing Email](#).

### Playbook Tasks

To define auto extract for a playbook task, do the following:

1. Select the playbook you want to add auto extract, and click **Edit**.
2. In the playbook, click a task to open the **Edit Task** window.
3. Click the **Advanced** tab.
4. In the auto extract drop down menu, select the mode you want to use.
5. Click **OK**.

### Cortex XSOAR CLI

To define auto extract using the Cortex XSOAR CLI, use the **auto-extract=** parameter with the script and the mode for which you are setting up auto-extract. For example, **!EmailReputation email=email@email.com auto-extract=inline**, filling in the script and mode you want to define.

## Configure What Auto Extract Executes

When Auto Extract is used, it extracts all indicators that match the regex defined in an indicator type, and enriches those indicators using its commands. For example, out-of-the-box, the URL indicator is enriched using the **!url** command. You can decide to further enrich IP indicators by using a script that calls multiple integrations, such as urlscan.io and URLhaus.



*By design, domains are extracted only from URLs and email addresses. Otherwise, the amount of incorrect extractions would be huge and every <text>.<text> would be considered as a domain indicator. So, for example, google.com will not be extracted, but https://google.com will.*

**STEP 1** | Navigate to **Settings > Advanced > Indicator Types**.

---

**STEP 2** | Select the indicator type for which you want to configure the command or script and click **Edit**.

For out of the box indicators, the Name and Regex fields are disabled.

**STEP 3** | Under **Reputation command**, enter the command to execute when auto extracting indicators of this type.

**STEP 4** | Under **Exclude these integrations for the reputation command**, select which integrations should not be used when executing the reputation command.

**STEP 5** | Under **Reputation Script**, select the script to run when enriching indicators of this indicator type. The scripts override the reputation command.

**STEP 6** | Click **Save**.

## Disable Auto Extract for Scripts and Integrations

You can disable auto-extract for a specific automation or integration.

**STEP 1** | Disable for an Automation.

To disable Auto Extract for an automation, add the `'IgnoreAutoExtract': True` value to the entry return.

```
entry = {
 'Type': entryTypes['note'],
 'Contents': { 'Echo' : demisto.args()['echo'] },
 'ContentsFormat': formats['json'],
 'ReadableContentsFormat': formats['markdown'],
 'HumanReadable': hr,
 'IgnoreAutoExtract' : True
}
```

**STEP 2** | Disable for an Integration.

To disable Auto Extract for a specific integration, add the `'IgnoreAutoExtract': True` value to the integration configuration.

## Auto Extract Indicators from a Phishing Email

The following scenario shows how Auto Extract is used in the Process Email - Generic playbook to automatically extract and enrich a very specific group of indicators.

**STEP 1** | Navigate to the **Playbooks** page and search for the **Process Email - Generic** playbook.

This playbook parses the headers in the original email used in a phishing attack. It is important to parse the original email used in the Phishing attack and not the email that was forwarded to make sure that you are only extracting and enriching the email headers from the malicious email and not the one your organization uses to report phishing attacks.

**STEP 2** | Open the **Add original email attachments to context** task.

Under the **Outputs** tab you can see all of the different data that the task extracts.

🔒 Edit Task: Add original email attachments to context ✕

ParseEmailFiles ▼ ?

Inputs
Outputs
Field mapping
Advanced
Details
Timers

**Email**

To This shows to whom the message was addressed, but may not contain the recipient's address.

CC Email 'cc' addresses

From This displays who the message is from, however, this can be easily forged and can be the least reliable.

Subject Email subject

HTML Email 'html' body if exists

Text Email 'text' body if exists

Depth The depth of the email. Depth=0 for the first level email. If email1 contains email2 contains email3. Then email1 depth is 0, email2 depth is 1, email3 depth is 2

Headers Deprecated - use Email.HeadersMap output instead. The full email headers as a single string

HeadersMap The full email headers json

AttachmentNames The list of attachment names in the email

Format The format of the email if available

**Email.HeadersMap**

From This displays who the message is from, however, this can be easily forged and can be the least reliable.

To This shows to whom the message was addressed, but may not contain the recipient's address.

Subject Email subject

Date The date and time the email message was composed

Stop on errors NOYES

I've seen enough

**STEP 3** | Navigate to the **Advanced** tab.

Under **Auto extract indicators**, ensure that the **InLine** option is selected. This indicates that all of the outputs will be processed before the playbook moves ahead to the next task.

**Edit Task: Add original email attachments to context** [X]

Inputs   Outputs   Field mapping   **Advanced**   Details   Timers

Using [?] Start typing and press enter to add instance. Leave empty to use all instances.

Extend context [?]

Ignore outputs [?]

Execution timeout (seconds) [?]

Number of retries [?]   Retry interval (seconds) [?]

Default is 0 (no retries)   Default is 30 Seconds

**Auto extract indicators** **Inline** [?] [v]

Mark results as note

Mark results as evidence

Run without a worker

Skip this branch if this automation/playbook is unavailable

Quiet Mode **Use playbook default** [?]

**Auto extracts and enriches all of the outputs before moving on to the next task**

**STEP 4** | Open the **Set incident with the Email object data** task. This task receives the data from the **Add original email attachments to context** task and sets the various data points to context.

Under the **Advanced** tab, ensure that **Auto extract indicators** is set to **None** because the indicators have already been enriched and there is no need to do it again.

### Edit Task: Set incident with the Email object data ✕

Standard  Conditional  Data Collection  Section Header

Task Name <sup>\*</sup>

Set incident with the Email object data

setIncident (Builtin) ?

Inputs   Outputs   Field mapping   **Advanced**   Details   Timers

Using ?

Start typing and press enter to add instance. Leave empty to use all instances.

Extend context ?

Ignore outputs ?

Execution timeout (seconds) ?

Number of retries ?   Retry interval (seconds)

Default is 0 (no retries)   Default is 30 Seconds

**Auto extract indicators** None ? 

Mark results as note

**Auto extract is disabled**

In the above example, had we set the reputation.calc.algorithm.tasks server configuration to 1, we would not have had to go into the Advanced tab of the Set incident with the Email object data task and manually tell the task not to extract the indicators. It would use the system default.

# Incidents

- > Incident Lifecycle
- > Incidents Management
- > Fetch Incidents from an Integration Instance
- > Classification and Mapping
- > Create a Search Query for Incidents
- > Create a Widget From an Incident
- > Customize Incident View Layouts
- > Incident Investigation
- > War Room Overview
- > Link Incidents
- > Investigate Using the Canvas
- > Incident Actions
- > Incident Tasks
- > Customize Incident View Layouts
- > Incident Fields
- > Incident De-Duplication
- > Create Pre-Process Rules for Incidents
- > Post Processing for Incidents
- > Incident Access Control Configuration



# Incident Lifecycle

Cortex XSOAR is an orchestration and automation system used to bring all of the various pieces of your security apparatus together.

Using Cortex XSOAR, you can define integrations with your 3rd-party security and incident management vendors. You can then trigger events from these integrations that become incidents in Cortex XSOAR. Once the incidents are created, you can run playbooks on these incidents to enrich them with information from other products in your system, which helps you complete the picture.

In most cases, you can use rules and automation to determine if an incident requires further investigation or can be closed based on the findings. This enables your analysts to focus on the minority of incidents that require further investigation.

The following diagram explains the incident lifecycle in Cortex XSOAR.



## Planning

Before you begin configuring integrations and ingesting information from 3rd parties, you should plan ahead.

| Phase                   | Description                                                                                                                                                                             |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create fields           | Used to display information from 3rd-party integrations and playbook tasks when an incident is created or processed. For more information, see <a href="#">Incident Fields</a> .        |
| Create incident types   | Classify the different types of attacks with which your organization deals.                                                                                                             |
| Create incident layouts | Customize your layouts for each incident type to make sure the most relevant information is shown for each type. For more information, see <a href="#">Customize Incident Layouts</a> . |

---

This is an iterative process. After you initially create your fields and incident types, as well as implement them in your incident layouts, you can start the process of ingesting information. You can then see how accurately you have mapped out your information. Make changes as you go along and learn more about the information you are receiving. Information that is not mapped to fields is available in labels, but it is much easier to work with the information when it is properly mapped to a field and displayed in the relevant layouts.

## Configure Integrations

You configure integrations with your 3rd-party products to start fetching events. Events can be potential phishing emails, authentication attempts, SIEM events, and more.

## Classification Mapping

Once you configure the integrations, you have to determine how the events ingested from those integrations will be classified as incidents. For example, for email integrations, you might want to classify items based on the subject field, but for SIEM events, you will classify by event type. In addition, you have to map the information coming from the integrations into the fields that you created in the planning stage. For more information, see [Classification and Mapping](#).

## Pre-Processing

Pre-processing rules enable you to perform certain actions on incidents as they are ingested into Cortex XSOAR directly from the UI. Using the rules, you can select incoming events on which to perform actions, for example, link the incoming event to an existing incident, or based on configured conditions, drop the incoming incident altogether. For more information, see [Create Pre-Process Rules for Incidents](#).

## Incident Created

Based on the definitions you provided in the Classification and Mapping stage, as well as the rules you created for pre-processing events, incidents of various types are created. The incidents all appear in the Incidents page of the Cortex XSOAR user interface, where you can start the process of investigating

## Running Playbooks

Playbooks are triggered either when an incident is created or when you run them manually as part of an investigation. When triggered as part of an incident that was created, the playbooks for the type of incident that was classified will run on the incident. Alternatively, if you are manually running a playbook, you can select whichever playbook is relevant for the investigation. For example, playbooks can take IP address information from one integration and enrich that IP address with information from additional integrations or sources.

## Post-Processing

Once the incident is complete and you are ready to close it out, you can run various post-processing actions on the incident. For example, send an email to the person who opened the incident informing them that their incident has been resolved, or close an incident in a ticketing system.

---

# Incidents Management

Incidents are events that have been observed at a point in time for analysis. Cortex XSOAR ingests incidents from an integration instance, from the REST API, or from an incident that you create manually.



To view the REST API documentation, select [Settings > INTEGRATIONS > API Keys > View Cortex XSOAR API](#).

In the Incidents page, you can view the following:

- All incidents in Cortex XSOAR.

By default, the Incidents page displays all open incidents from the last seven days. You can update this by creating a new [search query](#).

You can [Create a Widget From an Incident](#), based on the search query and add it to a dashboard or report.

- Incident categories in a bar chart format. You can change these categories by selecting a different chart from the drop down list from each individual chart. You can also hide the chart panel.
- All Incidents that are ingested into Cortex XSOAR in a table format, which is used to assign incidents, and perform batch actions on multiple incidents. You can see general information about the incident such as the type, the severity, when it occurred, and so on. The status of the incident is classified as follows:

**Active:** The investigation has started. The War Room is activated and the Playbook starts, if assigned. Users can be assigned to this incident.

**Pending:** The investigation has not started and no War Room has been activated. As soon as you open the incident, it becomes active.

**Closed:** The investigation has been closed.

You can limit access to investigations and restrict investigations according to your requirements, as described in [Incident Access Control Configuration](#).

When selecting the incident, you can do the following:

- **Investigate an incident:** You can view a detailed summary, investigate, add evidence, see related incidents and so on.
- **Assign:** You can assign incidents to any user that has been added to Cortex XSOAR.
- **Edit:** You can edit the incident parameters and then rerun the incident again, which is useful while developing playbooks. You can process an incident multiple times for a playbook while being developed, without creating new incidents every time.
- **Mark as Duplicate**
- **Run Command**
- **Export** to a CSV file
- **Close** the incident
- **Delete** the incident

You can create a new incident by clicking **New Incident**. You can also create a new incident in the REST API

You can filter the incidents that are ingested into Cortex XSOAR by [Manually De-Duplicate Incidents](#), setting up [pre-process rules](#) to perform certain actions, or [automatically de-duplicate incidents](#). After you close an incident you may want to automate an additional action such as closing a Remedy ticket. For more information, see [Post Processing for Incidents](#).

---

## Fetch Incidents from an Integration Instance

You can poll third party integration instances for events and turn them into Cortex XSOAR incidents that trigger automations (fetching).

There a number of integrations that support fetching, but not all support this feature. You can view each integration in the [Demisto Developer Hub](#).

You can set an integration to fetch events, when defining an integration from the **INTEGRATIONS** tab in the Settings page, by selecting the **Fetches incidents** check box.

Once enabled, Cortex XSOAR searches for events that occurred within the time frame set for the integration, which is based on the specific integration. The default is 10 minutes prior, but can be changed in the integration script implementation.

The next fetch depends on the “systemwide interval”. The default is 1 minute, but it is possible to override this by setting server configuration `server siem incidents schedule`. The value is the interval in seconds (s), minutes (m) or hours (h). You add a server configuration in **Settings > About > Troubleshooting**. For example, type `jobs.serversiemincidents.schedule` key and `120s` value.



*If you turn off fetching for a period of time and then turn it on or disabled the instance and enabled it, the instance remembers the "last run" timestamp, and pull all events that occurred while it was off. If you don't want this to happen, verify that the instance is enabled and then click Reset the "last run" timestamp in the settings window. Also, note that "last run" is retained when an instance is renamed.*

You set the objects to be fetched and their mapping in **Settings > INTEGRATIONS > Classification & Mapping**.

## Classification and Mapping

The classification and mapping feature enables you to take the events and event information that Cortex XSOAR ingests from integrations or REST API, and classify the event as a type of Cortex XSOAR incident.

For example, Cortex XSOAR might generate alerts from Traps which you would classify according to the information in those either as dedicated Traps incident types, Authentication or Malware. You might have EWS configured to ingest both phishing and malware alerts, which you would want to classify to their respective incident types based on some information in the event. By classifying the events differently, you have more control of the incident type and allowing you to run multiple playbooks for the events coming from one source.

### Classification

Classification determines the type of incident that is created for events ingested from a specific integration. You can classify events in the following ways:

- **Defining an integration**  
Select the incident type that is created. When this is configured, it becomes the default incident type. If you do not classify the event through classification and mapping, it is set as what you have defined here.
- **Setting a classification key**  
Use the classification engine to determine the incident type. This overrides whatever you configured in the integration settings.

---

## Mapping

Once you classify the incident, you can map the fields from the 3rd party integration to the fields that you defined in the incident layout. Any fields that you do not map, are automatically mapped to Cortex XSOAR labels. While this information can still be accessed, it is always easier to work with fields.

To get the most benefit out of classification and mapping, ensure that you understand which information is ingested from the events, so you can set up the fields and incident types accordingly.

### Classify Events Using a Classification Key

When an integration fetches incidents, it populates the `rawJSON` object in the incident object. The `rawJSON` object contains all of the attributes for the event. For example, source, when the event was created, the priority that was designated by the integration, and more. When classifying the event, you want to select an attribute that can determine what the event type is.

**STEP 1 |** Open the Classification & Mapping window for the Integrations instance.

1. Go to **Settings > Integrations > Servers & Services** and next to the integration instance, click **Mapping**
2. In the **Classification & Mapping** tab, from the dropdown menu, select the integration instance.

**STEP 2 |** In the **Values to Identify** column, drag values from the **Unmapped Values** column or type your own value.

**STEP 3 |** Click **Create mapping** to open the classification wizard.

**STEP 4 |** Load event data using one of the following options:

- **Pull events:** pull from `integrationName`. Cortex XSOAR fetches events from the instance (alerts, notifications etc.)
- **Pull events:** upload a JSON file containing the `rawJSON` object from the integration. The file must be uploaded in JSON format.
- **Skip getting samples:** Map the attributes without event data. Not recommended.

**STEP 5 |** Set the classification key.

The event attributes are presented on the right side of the screen. Click on the attribute by which you want to classify the incidents. You can navigate between the fetched events to view all of the attributes in the other events and to ensure that you are selecting a viable attribute.

You can use filters and transformers to make the selection more exact.

**STEP 6 |** Click **Done**.

Once you select the attribute, the unique values for the attribute that you have selected from the fetched events appear under the **Unmapped Values** column.

**STEP 7 |** Drag any unmapped value to the **Values to Identify** column for the incident type to which you want to classify. Any unmapped values that you do not classify, an incident type as defined in the integration is created.

You can map multiple values to an incident type, but you cannot map an unmapped value to multiple incident types.

**STEP 8 |** [Map Event Attributes to Fields](#) so the information is indexed.

---

## Map Event Attributes to Fields

You should map event attributes to the incident fields so the information is indexed. By default, attributes are not mapped to any fields. They are only available in the incident.labels of the incident.

### Before you begin

Ensure you have [classified events using the classification key](#).

**STEP 1** | In the incident type, click **Edit Mapping**

**STEP 2** | In the Mapping Wizard, in the **Map to** column, click **Choose data path**.

**STEP 3** | Click the event attribute to which you want to map. You can further manipulate the field using filters and transformers.

**STEP 4** | Click **Done**.

## Receive Notification on an Incident Fetch Error

The administrator and Cortex XSOAR users on the recipient's list receive a notification when an integration experiences an incident fetch error. Administrators with multiple instances of mail sender can choose to receive one email notification instead of multiple email notifications. Cortex XSOAR users can select their notification method, such as email, or mobile, from their user preferences.



*The connectivity behavior that exists between third-party applications may trigger a fetch failure, which will send a notification to an administrator and users. The notification may no longer be relevant because the fetch might operate correctly just after the notification was sent.*

### Before you begin

In the integration instance, ensure that you select the **Fetch Incidents** checkbox.

**STEP 1** | Select **Settings > About > Troubleshooting > Add Server Configuration**.

**STEP 2** | Add the following keys and values:

| Key                                              | Value                                                                     |
|--------------------------------------------------|---------------------------------------------------------------------------|
| <code>module.health.notification.users</code>    | List of names in CSV format, for example <code>user1,user2,user3</code> . |
| <code>message.ignore.failedFetchIncidents</code> | <code>false</code> .                                                      |

**STEP 3** | (Optional) Administrators that have multiple instances of mail sender configured that want to receive only one email notification need to add the following key and value:

| Key                                              | Value                          |
|--------------------------------------------------|--------------------------------|
| <code>server.notification.using.send-mail</code> | The mail-sender instance name. |

## Create a Search Query for Incidents

The default view of the Incidents page displays all open incidents from the last seven days. You can customize which incidents are displayed by creating and saving queries. You can also customize the information that is displayed for each incident by customizing the table summary layout and the Chart panel. This information is then saved as part of the query.

**STEP 1** | In the query bar, type your search criteria.

By default, the syntax is `-status:closed -category:job`, which searches for categories other than jobs and not those that have been closed. You can add fields like severity or type to narrow your search to critical issues or issues of a certain type.

**STEP 2** | From the drop down list, select the date range for which you want to search.

By default, it is the last 7 days.

**STEP 3** | If you want to customize the table summary view, click the gear icon above the table.

**STEP 4** | If you want to customize the chart panel, go to one of the charts and from the drop down list select the chart as required.

**STEP 5** | To save the query do the following:

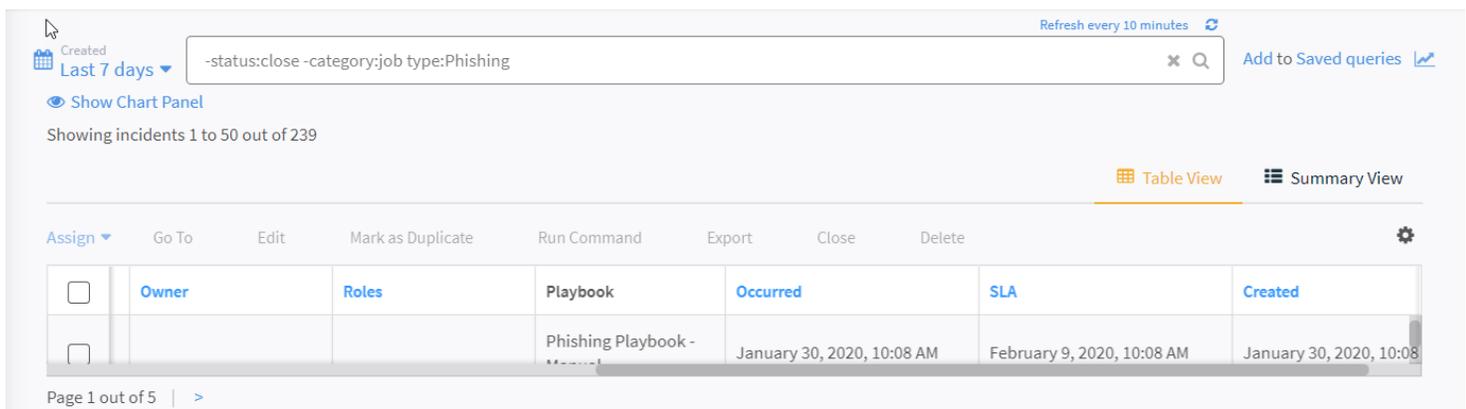
1. Click **Add** to Saved queries.
2. Type a name for the query.
3. Click **Save**.

When clicking **Saved queries** you can view all saved queries, mark them as default, or delete the queries.

In this example, you need to search for all incidents according to the following criteria:

- Status is not closed
- category is not job
- type is phishing
- opened within the last 7 days

In addition, add the **Created** column to the table summary.



The screenshot shows the Cortex XSOAR interface. At the top, there is a search bar with the query `-status:close -category:job type:Phishing`. To the left of the search bar, there is a date range selector set to "Last 7 days" and a "Show Chart Panel" button. To the right of the search bar, there is a "Refresh every 10 minutes" button and an "Add to Saved queries" button. Below the search bar, it says "Showing incidents 1 to 50 out of 239". At the bottom of the search bar area, there are two view options: "Table View" (selected) and "Summary View". Below this, there is a table with columns: "Assign", "Go To", "Edit", "Mark as Duplicate", "Run Command", "Export", "Close", "Delete", and a gear icon. The table has a header row with columns: "Owner", "Roles", "Playbook", "Occurred", "SLA", and "Created". The first row of data shows: "Phishing Playbook -", "January 30, 2020, 10:08 AM", "February 9, 2020, 10:08 AM", and "January 30, 2020, 10:08". At the bottom left, it says "Page 1 out of 5 | >".

---

## Create a Widget From an Incident

Although there are various out-of-the-box system widgets available, you can create custom widgets from incidents and then add them to a dashboard or report. You can also [create a widget in the widgets library](#).

To create a widget from an incident, you need to run a query from the Incidents page and then save the visual results as a widget. For an example, see [Create a Widget From an Incident Example](#).

**STEP 1** | In the **Incidents** page, from the drop down list select the date range.

**STEP 2** | In the query field, type the query criteria as required and run the query.

**STEP 3** | Click **Create a Widget** button.

**STEP 4** | Add the [Widget Parameters](#).

**STEP 5** | Click **Save**.

The widget is added to the **Widgets Library**.



*By default, the widget inherits the date range that you specify when creating the widget, but you can modify the date range when you create the dashboard or report. If the date range for the report or dashboard does not include the widget date range, the data is blank. To override the dashboard or report's date range, click Use Widget's date range.*

### Create a Widget From an Incident Example

In the following example, you need to create a widget that contains:

- Incidents created in the last 6 months
- Status: All statuses other than closed
- Category: All categories other than jobs
- Use Access Investigation - Generic playbook

**STEP 1** | In the Incident's page, run the following query:

Created  
Last 7 days ▾ -status:closed -category:job and playbook:"Access Investigation - Generic"

**STEP 2** | Click type the name (Closed Job Incidents with Access Investigation (past 6 months)) and save the query results as a widget:

Quick chart definitions ×

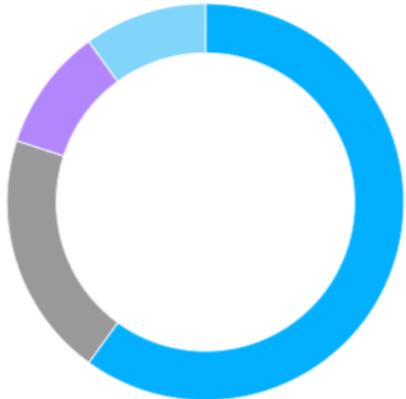
Data query:  × Q

Time frame: 📅 Last 6 months ▾ 🔔 Incidents by: Type ▾

📊 📋 📄 🕒 📏 📱 📄

Closed Job Incidents with Access Investigation (past 6 n) ⚙️

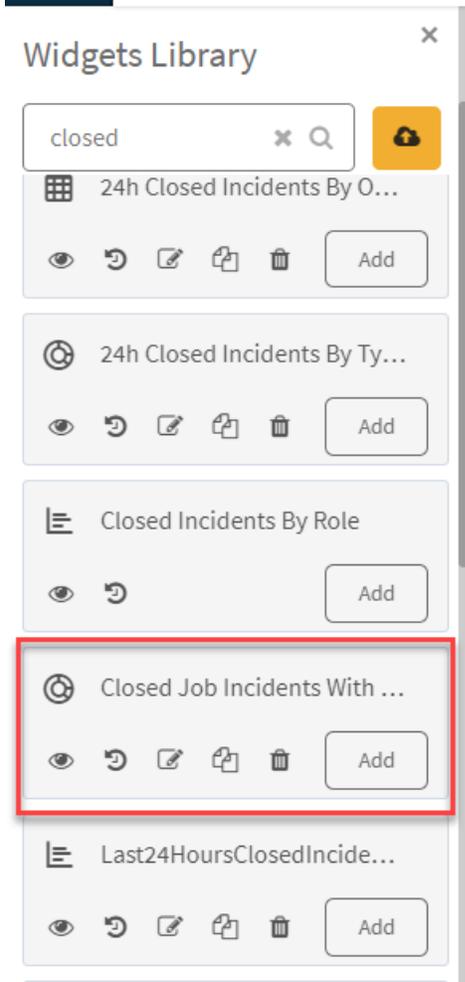
|                 |   |
|-----------------|---|
| ● Unclassified  | 6 |
| ● DESIGN        | 2 |
| ● Access        | 1 |
| ● Vulnerability | 1 |



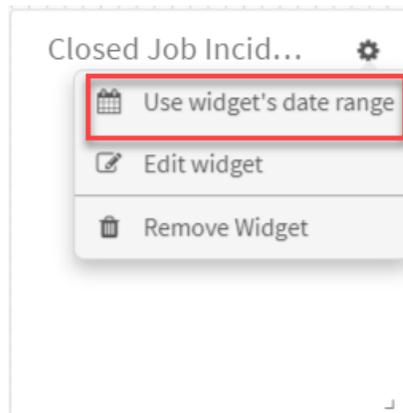
Cancel Save

STEP 3 | Add/Edit a dashboard and locate the widget:

# Dashboards



**STEP 4 |** Add the widget to the dashboard. If no data is returned, click **Use widget's date range**.



## Customize Incident View Layouts

Each incident type has a unique set of data relevant to that specific incident type. It is important to display the most relevant data for users at all stages of the incident life cycle.

You can customize the display information for existing incidents, and the fields in incident forms, by modifying the sections and fields for each view.



*There are several Cortex XSOAR system layout sections and fields that you cannot remove, but you can rearrange them in the layout and modify their queries and filters.*

**STEP 1** | Go to **Settings > Advanced > Incident Types**.

**STEP 2** | Select the incident type check box that you want to customize.

**STEP 3** | Click **Edit Layout**.

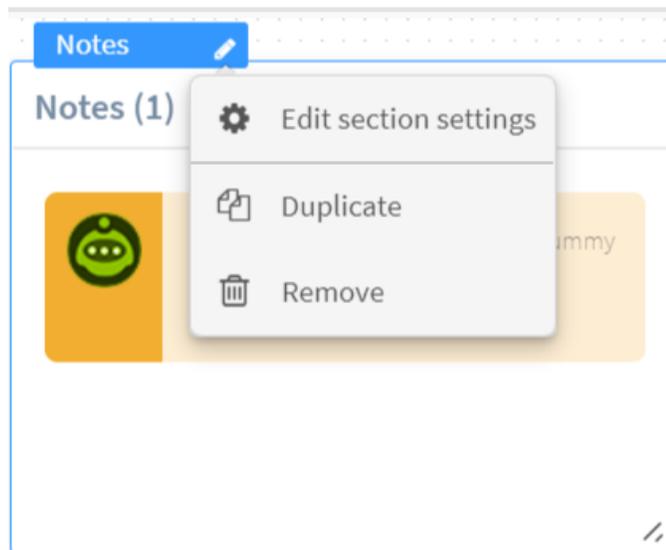
**STEP 4** | Select one or more of the following views to edit:

| View                | Description                                                                                     |
|---------------------|-------------------------------------------------------------------------------------------------|
| Incident Summary    | The fields and sections displayed in the <b>Incident</b> page.                                  |
| “New”/“Edit Form    | The fields and sections displayed when creating a new incident or editing an existing incident. |
| “Close” Form        | The fields and sections when closing an incident.                                               |
| Incident Quick View | The fields and sections when displaying the incident quick view.                                |
| Mobile              | The fields or sections displayed on a mobile.                                                   |

**STEP 5** | From the **Library** section, in the **Cortex XSOAR Sections** drag and drop the following sections as required.

| Section                              | Description                                                                                                                                                                                                                 |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| New Section                          | After creating a new section, click the <b>Access Fields</b> tab and drag and drop the fields as required.                                                                                                                  |
| Cortex XSOAR out of the box sections | Out of the box sections such as Attachments, Evidence, and so on.                                                                                                                                                           |
| General Purpose Dynamic Section      | Enables you to assign a script to this section. For example, assign a script that calculates the total number of entries that exist for an incident, and it dynamically updates when new entries are added to the incident. |

**STEP 6 |** (Optional) To edit the section, select the section, click  and then **Edit section settings**.



**STEP 7 |** Edit the section as required and click **OK**.

**STEP 8 |** Click the save button or **Save Version**.

## *Customize Incident Layouts*

After configuring your fields and incident types, it is important to build or customize the layout to ensure that you are seeing the information that is germane to the incident type. For example, for a phishing incident you will want to see email headers, which would not be relevant for an access incident. Additionally, while some information might be relevant for multiple incident types, its location in one incident type might require more prominence than in another incident type.

You can customize almost every aspects of the layout, including, but not limited to:

- which tabs appear
- in which order do they appear
- who has permissions to view the tabs
- which information appears and how is it displayed

You can add dynamic sections to a layout, such as a graph of the number of bad indicators, their source, and severity. Also, you can use queries to filter the information in the dynamic section to suit your exact needs.

## *Customize an Incident Type Layout*

Each Incident Type should show the information that is most relevant to it. That includes the right fields and information types in the right places.

**STEP 1 |** Navigate to **Settings > Advanced > Incident Types**.

**STEP 2 |** Select the incident type whose layout you want to edit and click **Edit Layout**.

You are presented with the current layout, which is populated with demo data so you can see how the fields fit.

**STEP 3 | (Optional)** Drag and drop the tabs to reorder their appearance. For example, drag the War Room so it appears after the Work Plan. You can also click **+New tab** to add a tab that currently does not exist.

**STEP 4 |** Manage general settings for a tab.

You can configure which tabs appear and for whom, as well as duplicate or remove tabs from the layout.

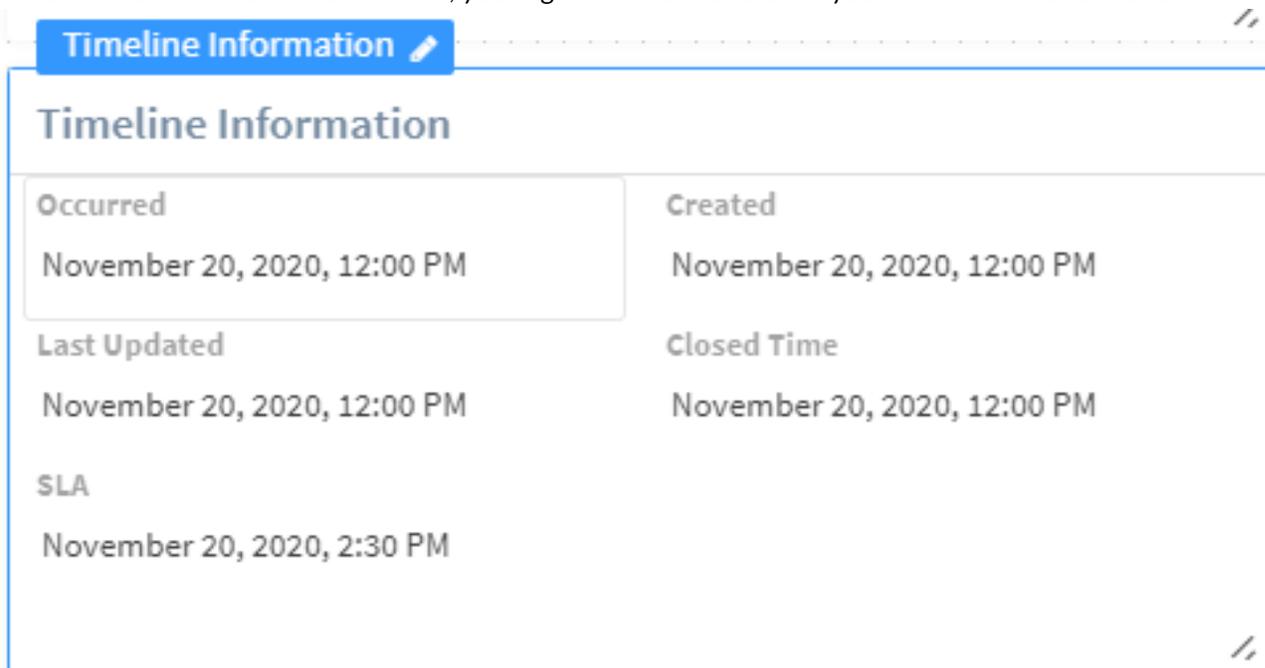
1. Hover over the tab that you want to configure.
2. Click the gear icon.

You are presented with the following options:

- Rename
- Duplicate
- Delete
- Hide
- Viewing Permissions

**STEP 5 |** Define the section properties.

You can determine how a section in the layout appears in the layout. For example, does the section include the section header or not. You can also configure the fields to appear in rows or as cards. For example, if you know that some of the field values will be very long, you are better off using rows. If you know that the field values are short, you might want to use cards so you can fit more fields in a section.



1. To remove or duplicate a section, or change its properties:
  - Click the section title. In the image above, that is Timeline Information.
  - Click the pencil icon and select the relevant option.
2. To change the information that appears in dynamic sections:
  - Click the section title. For example, Indicators.
  - Click the pencil icon and select **Edit section settings**.
  - Under **Query**, enter the parameters by which you want to filter the information that appears.

For example, to see all indicators of type IP and with a reputation of Bad that were found by a specific feed since March 1st 2020, enter Type:IP and reputation:Bad and firstseenbyfeed:>="2020-03-01T00:00:00 +0200".

- Click **OK**.

## STEP 6 | Add New sections or fields to a Layout

You can add new sections or fields to the layout by dragging and dropping them from the Library on the left into the layout.

### Example - Add a Button Field

1. Under the Fields section of the Layout Builder Library, drag the **+New Button** field into the relevant section.
2. Click the **Click to configure** button.

The screenshot shows the Cortex XSOAR interface. On the left, the 'Authentication Fields' library is visible, with a '+ New Button' field highlighted by a dashed blue box and a hand cursor. Below it are 'Attachment', 'Category', and 'Close Notes' fields. On the right, the 'Incident Info' page is shown, with the 'Case Details' tab selected. The 'Case Details' table contains the following information:

| Case Details    |               |
|-----------------|---------------|
| Type            | Incident type |
| Severity        | Low           |
| Owner           | johndoe       |
| Source          | Source        |
| Source Brand    | Brand name    |
| Source Instance | Instance type |
| Playbook        | Playbook name |

3. Enter a descriptive name for the button and select the script that you want to run when the button is clicked.
4. Click **Save**.

## Add a Custom Widget to the Incident page

You can add a custom widget in the **Case info** and **Investigation** tabs when you view a detailed incident. Every time you open the detailed incident you can see the widget.

Before you start, you need to create a custom widget, as described in [Create a Custom Widget Using an Automation Script](#)

 *In the Automation page, when adding or editing the script you want to use, ensure that you add the dynamic-section label.*

## STEP 1 | Go to **Settings** > **Advanced**.

**STEP 2** | To add a widget to an incident, click the **Incident Types** tab.

**STEP 3** | Select the incident type in which you want the widget to appear and click **Edit layout**.

**STEP 4** | From the **Layout Builder** window, in the **Library** section, drag the **General Purpose Dynamic Section** into the layout area in which you want it to appear.

**STEP 5** | In the **General Purpose Dynamic Section**, click the edit button.

**STEP 6** | Type a name and description for the widget.

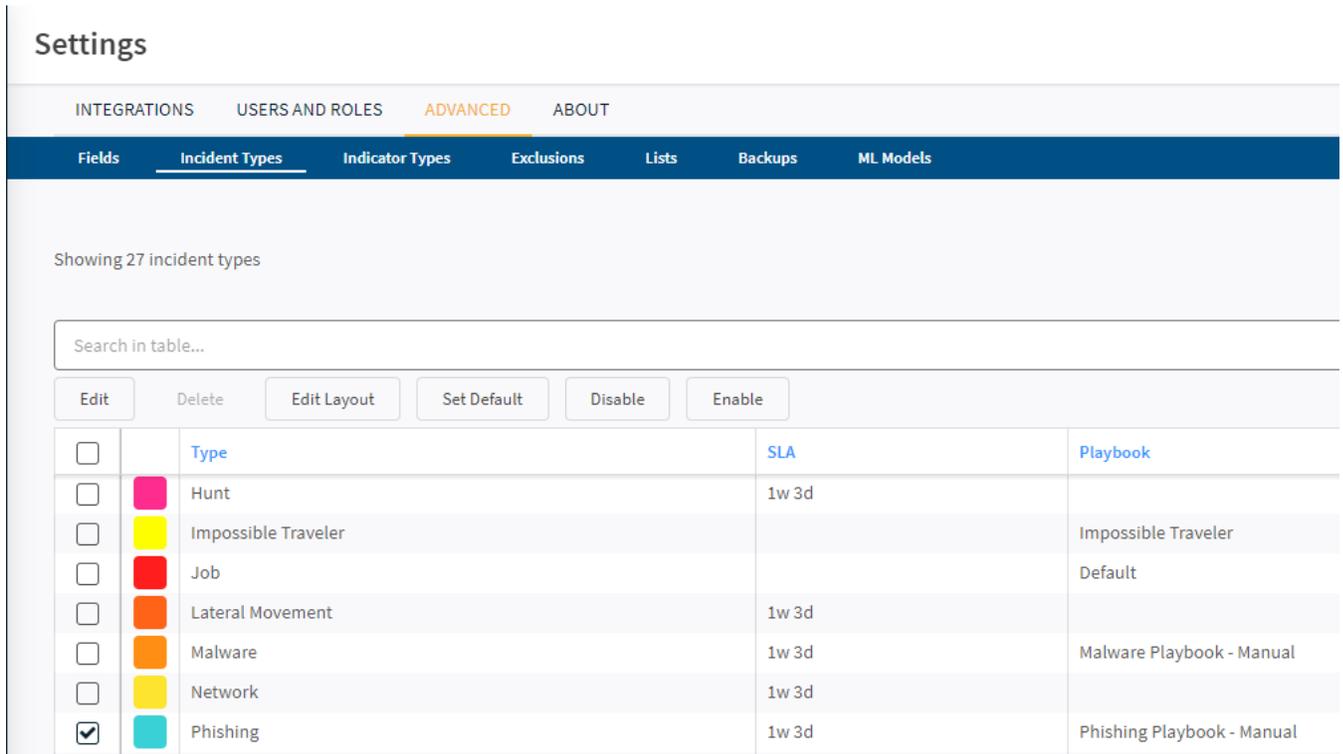
**STEP 7** | In the **Automation script** field, from the drop down list select the automation script you want to add.

 *If the automation script does not appear, you need to add the dynamic-section label to the script in the Automation page.*

**STEP 8** | Click **OK**.

The following example, shows how to add the Indicator Widget Bar to all **Phishing** incident types in the **Case info** tab.

1. In the **Incidents Types** tab, select **Phishing**.
2. Click **Edit Layout**.



The screenshot shows the 'Settings' page with the 'Incident Types' tab selected. A table lists 27 incident types. The 'Phishing' incident type is selected, indicated by a checked checkbox and a blue highlight. The table columns are 'Type', 'SLA', and 'Playbook'.

| <input type="checkbox"/>            | Type                | SLA   | Playbook                   |
|-------------------------------------|---------------------|-------|----------------------------|
| <input type="checkbox"/>            | Hunt                | 1w 3d |                            |
| <input type="checkbox"/>            | Impossible Traveler |       | Impossible Traveler        |
| <input type="checkbox"/>            | Job                 |       | Default                    |
| <input type="checkbox"/>            | Lateral Movement    | 1w 3d |                            |
| <input type="checkbox"/>            | Malware             | 1w 3d | Malware Playbook - Manual  |
| <input type="checkbox"/>            | Network             | 1w 3d |                            |
| <input checked="" type="checkbox"/> | Phishing            | 1w 3d | Phishing Playbook - Manual |

3. After adding the **General Purpose Dynamic Section** into the layout area, edit the widget, by adding the name and script.

## Section Settings General Purpose D... x

Name \*

Indicator Widget Bar

Show section header

Description

Automation script:

IndicatorWidgetBar x ⓘ

Cancel

OK

4. Go to the **Incidents** page.
5. Select an incident with incident type **Phishing**.
6. Click the **Case Info** tab.

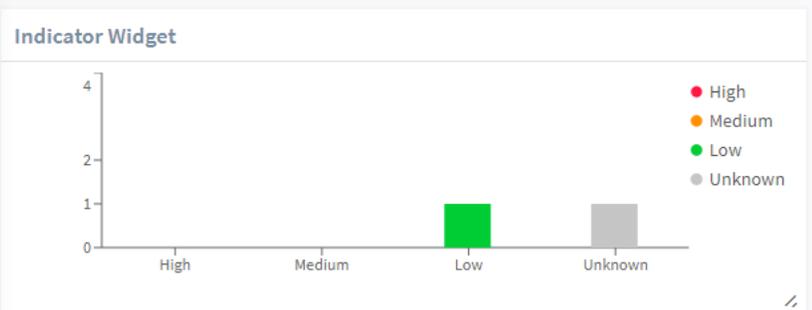
☆ #1257443 #72\_3 73739136-9027-4a70-8ec4-b4209cc9a519 - Case info

Summary **Case info** Investigation War Room Work Plan ▲ Evidence Board Related Incidents

### Case Details

|                 |                                         |
|-----------------|-----------------------------------------|
| Type            | Phishing                                |
| Source Instance | mail-listener_instance_6                |
| Severity        | <span style="color: red;">🔥</span> High |
| Source Brand    | mail-listener                           |
| Source          | N/A                                     |
| noa's field     | yes                                     |

### Indicator Widget



| Severity | Count |
|----------|-------|
| High     | 0     |
| Medium   | 0     |
| Low      | 1     |
| Unknown  | 1     |

Work Plan (2)  
Waiting for users (0)

Team (0)

---

## Add a Dynamic Section to an Incident Layout

You can add automation script-based content to the Incident Summary and Incident Quick View layout, by adding the **General Purpose Dynamic Section** in the layout builder.

The **General Purpose Dynamic Section** enables you to configure a section in the **Incident Info** tab from an automation script. The automation can return a simple text, markdown, or an HTML, the results of which appear in **General Purpose Dynamic Section**.

You can add any required information from an automation. For example, you can assign a script that calculates the total number of entries that exist for an incident, and it dynamically updates when new entries are added to the incident. You can also [Add a Custom Widget to the Incident page](#) and [Add Note Information Using an Automation Script](#).

Before you begin, you need to create an automation script.

**STEP 1** | Select the incident type you want to add the General Purpose Dynamic Section by completing steps 1 to 3 in [Customize Incident View Layouts](#).

**STEP 2** | In the **Incident Summary** tab, drag and drop the **General Purpose Dynamic Section** onto the page.

You can also select this section in the **Incident Quick View** tab.

**STEP 3** | Select the **General Purpose Dynamic Section**, click  and then **Edit section settings**.

**STEP 4** | In the **Name** and **Description** fields, add a meaningful name and a description for the dynamic section that explains what the script displays.

**STEP 5** | In the **Automation script** field, from the drop down list, select the script that returns data for the dynamic section.

For the script to appear, the script needs the `dynamic-section` tag assigned in the Automation page.

## Script settings



### Basic

Name

EntryWidgetNumberHostsXDR

Language type

Python ▾

Version

3.x ▾

Description

Entry widget that returns the number of hosts in a Cortex XDR incident.

Tags

dynamic-section

Enabled

**STEP 6** | Click **OK**.

## Add Note Information Using an Automation Script

This example shows how to add note information to an incident using an automation script through the API. For this script to run, you need to add a Demisto REST API instance. You add the script to the automation page and then add the script to the incident layout builder using the General Purpose Dynamic Section.

**STEP 1** | Go the Automation page and add the following script:

```
commonfields:
 id: ShowLastNoteUserAndDate
 version: -1
 name: ShowLastNoteUserAndDate
 script: |2

 function getLastNote(incidentID) {
 var body = {pageSize:1,categories:['notes']};
 var res = executeCommand('demisto-api-post', {uri:'/investigation/' +
incidentID, body: body});
 if (isError(res[0])) {
 throw 'demisto-api-post failed for incidnet #' + incidentID + '\nbody
is ' + JSON.stringify(body) + '\n' + JSON.stringify(res);
 }
 if (!res[0].Contents.response.entries) {
```

```

 return null;
 }
 var notes = res[0].Contents.response.entries;
 var lastNote = notes[notes.length-1];
 return lastNote;
}

lastNote = getLastNote(incidents[0].id);

if (lastNote) {
 md = `#### Update by ${lastNote.user} on
 ${lastNote.modified.split('T')[0]}\n`;
 md += `\n---\n`;
 md += lastNote.contents + '\n';

 return { ContentsFormat: formats.markdown, Type: entryTypes.note,
 Contents: md } ;
} else {
 return 'N/A';
}
}
type: javascript
tags:
- dynamic-section
enabled: true
scripttarget: 0
runonce: false
runas: DBotWeakRole

```

**STEP 2** | Select the incident type to add the script, by completing steps 1 to 4 in [Add a Dynamic Section to an Incident Layout](#).

**STEP 3** | In the **Automation script** field, select the automation added in step 1.

**STEP 4** | Go to the incident that you want to view the note information.

You can see note information, containing the last user and date.



## Create Dynamic Fields in Incident Forms

You can control which fields display in an incident form, and which values display for single-select and multi-select fields. The scripts support JavaScript and Python.

This can be useful in the following scenarios:

- You want to have certain values appear in a certain field when the value of another field is **a** or **b**.  
For example, if the value in the **Owner** field is **Admin**, the values in the assignee field should be **Jane**, **Joe**, or **Bob**. However, if the value in the **Owner** field is anything else, the values in the assignee field should be **Mark**, **Jack**, or **Christine**.
- You want a field to be available in a form only in certain conditions.

For example, if the value in the **Severity** field is **High** or **Critical**, the **Escalate** field should be available with values of **Yes** or **No**.

However, if the value in the **Severity** field is **Low** or **Informational**, the **Escalate** field should not appear.

**STEP 1 |** Navigate to the **Automation** page and duplicate the `hideFieldsOnNewIncident` automation.

1. Give the script a descriptive name.
2. Enter a useful description.
3. Under **Tags**, make sure that the `field-display` tag appears.

This tag must be applied for the script to be available to be used on the field.

4. Save the automation.

**STEP 2 |** Write the automation script.

This can be done either in the script window to the left, or in the script settings on the right. Review the examples at the end of this section for more information.

**STEP 3 |** Add the script to the relevant field.

1. Navigate to **Settings > Advanced > Fields**.
2. Click **+New Field** to create a field, or select the existing field to which you want to add the script and click **Edit**.
3. Under **Field Type**, select the field type. For example, Single select.
4. Under **Field Name**, enter a descriptive name.
5. Under the **Attributes** tab, in the **Field display script** field, select the script you want to run on this field.
6. Complete the remaining field-definitions and click **Save**.

**STEP 4 |** Implement the **Assign To** field in the relevant layouts. For more information, see [Customize an Incident Type Layout](#).

---

## Example - Change Field Values Dynamically

The following example shows how you would create a script for the Assignee field, which shows different values depending on the values in the **Owner** field. If the Owner is defined as 'admin', the list of available assignees will include one group of people. If the Owner is defined as anything else, the list of available assignees will include a different group of people.

1. In the **Automations** page, we copy the `hideFieldsOnNewIncident` and name it `changeAssigneesPerOwner`.
2. In the **Description** field, we enter the following:  
Changes values available in the Assignees field based on the person defined as the owner.
3. Under **Tags**, let's add the `field-display` tag.
4. For the automation, we enter the following code:

```
incident = demisto.incidents()[0]
field = demisto.args()['field']
if incident.get('owner') == 'admin':
 demisto.results({'hidden': False, 'options': ['jane','joe', 'bob']})
else:
 demisto.results({'hidden': False, 'options': ['mark','jack', 'christine']})
```

where

- `demisto.incidents` is the incident in which this script is running.
  - `incident.get('owner')` is the field within the incident.
  - `demisto.results` tells us whether to hide the field or not, and which values should appear in the field. When the owner is admin, the values will be Jane, Joe, and Bob. When the owner is anyone else, the values will be Mark, Jack, and Christine.
5. We navigate to **Settings > Advanced > Fields** and click **+New Field**.
    - We'll call the field **Assign To**.  
The **Values** field in the **Basic Settings** tab has been left blank because we hard-coded the values in our script.
    - Under the **Attributes** tab, in the **Field display script** field, select the `changeAssigneesPerOwner` script we created above.
    - Fill in the rest of the field definitions as desired and click **Save**.

6. We have to implement the field in an incident layout. For our example, we'll use the Authentication incident type.
7. Lastly, we create an incident and see what happens when the Owner is set to Admin and when the Owner is set to anything else.

### Example - Hide Field based on Context

In this example, we'll show how to hide a field for a new incident form, but display the field when editing the form. We'll also set field values for a multi-select field in the case of an existing incident.

For this example, we use the `hideFieldsOnNewIncident` out-of-the-box automation.

```

incident = demisto.incidents()[0]
field = demisto.args()['field']
formType = demisto.args()['formType']
if incident["id"] == "":
 # This is a new incident, hide the field
 demisto.results({"hidden": True, "options": []})
else:
 # This is an existing incident, we want to show the field, to know which
 values to display
 options = []
 # The field type includes the word select, such as Single select or Multi
 select
 if "Select" in demisto.get(field, "type"):
 # take the options from the field definition
 options = demisto.get(field, "selectValues")

```

```
demisto.results({"hidden": False, "options": options})
```

1. We navigate to **Settings > Advanced > Fields**.
2. Select the **Malicious Cause** field and click **Edit**.
3. Under the **Field display script** field, select the **hideFieldsOnNewIncident** script and click **Save**.
4. Navigate to the **Incidents** page and click **New Incident**.
5. Under the **Type** field, select **GDPR Data Breach**.

Scroll down and note that under **Mandatory Information**, there is no **Malicious Cause** field.

6. Click **Create New Incident** to save the incident.
7. Select the incident you just created and click **Edit**.

Scroll down to the **Mandatory Information** section and note that the **Malicious Cause** field appears and the options for the field are retrieved from the initial field definition.

Mandatory Information

Please fill in mandatory information about the suspected breach. Later, you can fill in more incident fields in the summary page based on your knowledge.

Company Name:  Contact Name:  DPO E-mail Address:

Country where the breach took place:  Date/time of the breach:  Choose time and date Affected data:

Approximate number of affected data subjects:  Data Encryption Status:  Where is data hosted:

Possible Cause of the Breach:  Malicious Cause (If the cause is a malicious attack):

Affected Individuals Contact Information:

| Full Name                              | E-mail Address                              |
|----------------------------------------|---------------------------------------------|
| <input type="text" value="bob james"/> | <input type="text" value="bob@sample.com"/> |

+ Add row

set form

Cancel Update Incident

## Examples of Script Based Widgets for Incident Layouts

The following are examples of the script based widgets that are supported in incident layouts:

- [Charts](#)
- [Pie](#)
- [Duration](#)
- [Number](#)
- [Number Trend](#)

### Charts

A valid result for a chart widget is a list of groups. Each group points to a single entity. For example, in bar charts each group is a bar. A group consists of the following:

- **Name** - A string.
- **Data** - An array of integers.
- **Color** - A string representing a color that will be used as a default color for that group. It can be the name of the color, a hexadecimal representation of the color, or an rgb color value (optional).

- **Groups** - A nested list of groups (optional).

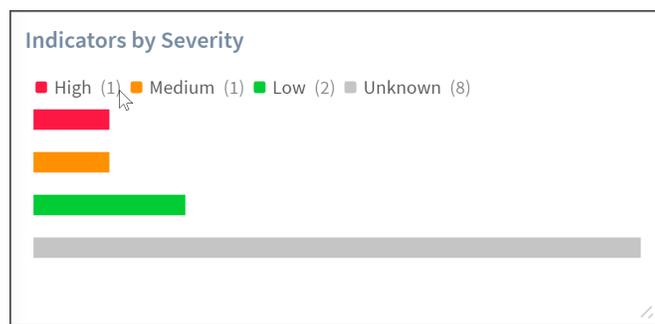
### Vertical Bar

In this example, create a script in Python that displays a vertical bar of the indicators by severity.

```
data = {
 "Type": 17,
 "ContentsFormat": "bar",
 "Contents": {
 "stats": [
 {
 "data": [
 1
],
 "groups": None,
 "name": "high",
 "label": "incident.severity.high",
 "color": "rgb(255, 23, 68)"
 },
 {
 "data": [
 1
],
 "groups": None,
 "name": "medium",
 "label": "incident.severity.medium",
 "color": "rgb(255, 144, 0)"
 },
 {
 "data": [
 2
],
 "groups": None,
 "name": "low",
 "label": "incident.severity.low",
 "color": "rgb(0, 205, 51)"
 },
 {
 "data": [
 8
],
 "groups": None,
 "name": "unknown",
 "label": "incident.severity.unknown",
 "color": "rgb(197, 197, 197)"
 }
],
 "params": {
 "layout": "vertical"
 }
 }
}

demisto.results(data)
```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



## Horizontal Bar

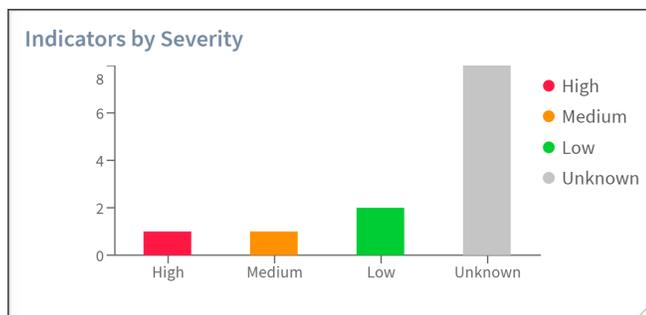
In this example, create a script in Python that displays a horizontal bar of the indicators by severity.

```
data = {
 "Type": 17,
 "ContentsFormat": "bar",
 "Contents": {
 "stats": [
 {
 "data": [
 1
],
 "groups": None,
 "name": "high",
 "label": "incident.severity.high",
 "color": "rgb(255, 23, 68)"
 },
 {
 "data": [
 1
],
 "groups": None,
 "name": "medium",
 "label": "incident.severity.medium",
 "color": "rgb(255, 144, 0)"
 },
 {
 "data": [
 2
],
 "groups": None,
 "name": "low",
 "label": "incident.severity.low",
 "color": "rgb(0, 205, 51)"
 },
 {
 "data": [
 8
],
 "groups": None,
 "name": "unknown",
 "label": "incident.severity.unknown",
 "color": "rgb(197, 197, 197)"
 }
],
 "params": {
 "layout": "horizontal"
 }
 }
}
```

```
}
}
```

```
demisto.results(data)
```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



### Stacked Bar

In this example, create a script in Python that displays a stacked bar showing the success and failures on specific dates.

```
data = {
 "Type": 17,
 "ContentsFormat": "bar",
 "Contents": {
 "stats": [
 {
 'name': 'time1',
 'groups': [
 {
 'name': 'Successes',
 'data': [7],
 'color': 'rgb(0, 205, 51)'
 },
 {
 'name': 'Failures',
 'data': [3],
 'color': 'rgb(255, 144, 0)'
 }
]
 },
 {
 'name': 'time2',
 'groups': [
 {
 'name': 'Successes',
 'data': [9],
 'color': 'rgb(0, 205, 51)'
 },
 {
 'name': 'Failures',
 'data': [4],
 'color': 'rgb(255, 144, 0)'
 }
]
 }
]
 }
}
```

```

 }
]
},
"params": {
 "layout": "horizontal"
}
}
}

```

`demisto.results(data)`

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



### Pie

In this example, create a script in Python that queries and returns a pie chart.

```

data = {
 "Type": 17,
 "ContentsFormat": "pie",
 "Contents": {
 "stats": [
 {
 "data": [
 1
],
 "groups": None,
 "name": "high",
 "label": "incident.severity.high",
 "color": "rgb(255, 23, 68)"
 },
 {
 "data": [
 1
],
 "groups": None,
 "name": "medium",
 "label": "incident.severity.medium",
 "color": "rgb(255, 144, 0)"
 },
 {
 "data": [
 2
],
 "groups": None,
 "name": "low",
 "label": "incident.severity.low",
 "color": "rgb(0, 205, 51)"
 }
]
 }
}

```

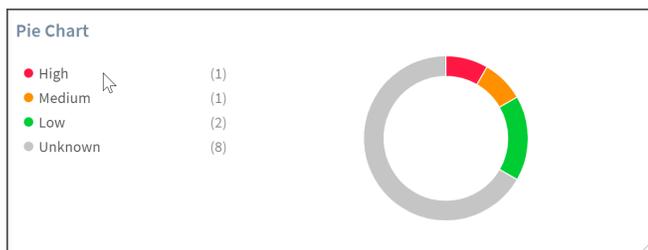
```

 },
 {
 "data": [
 8
],
 "groups": None,
 "name": "unknown",
 "label": "incident.severity.unknown",
 "color": "rgb(197, 197, 197)"
 }
],
 "params": {
 "layout": "horizontal"
 }
}

```

`demisto.results(data)`

After you have uploaded the script and created the widget, you can add the widget an incident layout. The following widget displays indicator severity as a pie chart:



### Duration

In this example, create a script in Python that queries and returns a time duration (specified in seconds), and displays the data as a countdown clock.

```

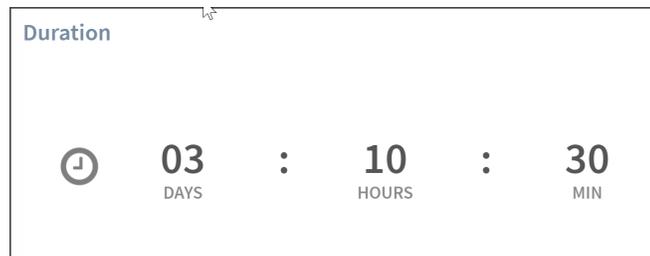
data = {
 "Type": 17,
 "ContentsFormat": "duration",
 "Contents": {
 "stats": 60 * (30 + 10 * 60 + 3 * 60 * 24),
 "params": {
 "layout": "horizontal",
 "name": "Lala",
 "sign": "@",
 "colors": {
 "items": {
 "#00CD33": {
 "value": 10
 },
 "#FAC100": {
 "value": 20
 },
 "green": {
 "value": 40
 }
 }
 }
 }
 },
 "type": "above"
}

```

```
}
```

```
demisto.results(data)
```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays the time duration:



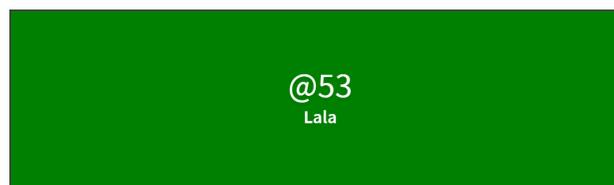
### Number

This example shows how to create a single item widget that displays a number.

```
data = {
 "Type": 17,
 "ContentsFormat": "number",
 "Contents": {
 "stats": 53,
 "params": {
 "layout": "horizontal",
 "name": "Lala",
 "sign": "@",
 "colors": {
 "items": {
 "#00CD33": {
 "value": 10
 },
 "#FAC100": {
 "value": 20
 },
 "green": {
 "value": 40
 }
 }
 }
 }
 },
 "type": "above"
}
```

```
demisto.results(data)
```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



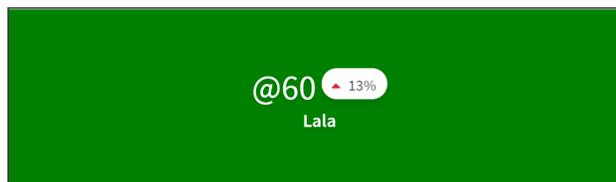
## Number Trend

This example shows how to create a single item widget that displays a number trend.

```
data = {
 "Type": 17,
 "ContentsFormat": "number",
 "Contents": {
 "stats": { "prevSum": 53, "currSum": 60 },
 "params": {
 "layout": "horizontal",
 "name": "Lala",
 "sign": "@",
 "colors": {
 "items": {
 "#00CD33": {
 "value": 10
 },
 "#FAC100": {
 "value": 20
 },
 "green": {
 "value": 40
 }
 }
 }
 }
 },
 "type": "above"
}

demisto.results(data)
```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



## Customize Incident Close Reasons

The default incident Close Reason values are:

- False Positive
- Resolved
- Duplicate
- Other

To customize the values you need to add a new server configuration.

**STEP 1** | Select **Settings** > **About** > **Troubleshooting** > **Add Server Configuration**.

**STEP 2** | Add the following key and value:

| Key                                | Value                                                                                                                                                                                                                                                                                                                           |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>incident.closeReasons</code> | <p>A comma-separated list. For example, <b>False Positive, Resolved, Duplicate, Low Priority, Invalid, Other</b></p> <p> <i>For multi-tenant deployments, you need to add the configuration to each tenant, not only the main account.</i></p> |

## Change the Display Name of Security Incidents

In Cortex XSOAR, the default term used for a security incident is *incident*. You can change the term that is used for security incidents from a predefined list of options. This term displays in reports, menus, tables, and commands (local and server) in Cortex XSOAR.

The term you select does not change the display name for content-related items, such as playbooks, integrations, scripts, and so on, dashboards, or the API.

 *(Multi-Tenant)* When changing the display name of security incidents, the URL link which contains `/incident` may not work properly. For example, when changing the **incident** to **case**, sometimes the links are formed with the `/incident` URL and not with the `/case` URL. This can usually be corrected by clearing the browser cache and reloading the page.

**STEP 1** | Navigate to **Settings > About > Troubleshooting**.

**STEP 2** | In the **Server Configuration** section, click **Add Server Configuration**.

**STEP 3** | In the **Key** field type `ui.term.incident`.

**STEP 4** | In the **Value** field enter the value for the term.

| Value | Term                |
|-------|---------------------|
| 0     | Incidents (default) |
| 1     | Cases               |
| 2     | Alerts              |
| 3     | Events              |
| 4     | Plays               |
| 5     | Tickets             |
| 6     | Issues              |

**STEP 5** | Restart the server.

# Incident Investigation

An incident investigation can be opened in the following ways:

- **Automatically:** If associated with a playbook, incidents open automatically for investigation and run the associated playbook.
- **Manually:** Open an incident manually by selecting the incident in the Incidents table.



After an incident is created, it is assigned a Pending status in the incident table. When you start to investigate an incident the status changes automatically to Active, which starts the remediation process.

- **CLI:** If you want to open an incident in the CLI, type `/investigate id=<incidentID#>`.

## Incidents page

When opening an incident, you see the following tabs, which assist you in the investigating the incident:

| Tab                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Incident/Case Info | <p>A summary of the incident, such as case details, work plan, evidence, and so on. Most of the fields are for information only, although you can add the following:</p> <ul style="list-style-type: none"><li>• <b>Evidence:</b> A summary of data marked as evidence. You can add evidence in this tab or in the <a href="#">Evidence Board</a>.</li><li>• <b>Notes:</b> Displays any notes that have been entered. For example, understand specific actions taken by the analyst and the underlying reasons, see chats between analysts to highlight how they arrived at a certain decision, etc. You can also see the thought process behind identifying key evidence and learn about similar incidents in the future.</li></ul> <p>You can also add notes in the War Room.</p> <ul style="list-style-type: none"><li>• <b>Tasks:</b> View tasks to complete as part of an investigation. You can add tasks in this tab or <a href="#">Create a To-Do Task</a>.</li></ul> <p>You can send a permalink to a specific Investigation Summary by copying its URL.</p> <p> You can edit the fields by <a href="#">customizing incident view layouts</a>.</p> |
| Investigation      | An overview of the information collected about the investigation, such as indicators, email information, URL screen shots and so on                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| War Room           | A comprehensive collection of all investigation actions, artifacts, and collaboration. It is a chronological journal of the incident investigation. Each incident has a unique War Room.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Work Plan          | A visual representation of the running playbook that is assigned to the incident.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Evidence Handling  | View any entity which has been designated as evidence. The Evidence board stores key artifacts for current and future analysis. You can reconstruct attack chains and piece together key pieces of verification for root cause discovery.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Related Incidents  | A visual representation of incidents that share similar characteristics, such as malicious indicators, or part of a phishing campaign.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

| Tab    | Description                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Canvas | <p>Visually maps an incident, its elements, correlated investigation entities, and the progression path of the incident, combining analyst intelligence with machine learning.</p> <p>The <b>Related Incidents</b> page is orientated towards exploration and searching for similar data. The <b>Canvas</b> maps incidents and indicators by enabling you to decide what you want to include in a layout of your choice.</p> |

You can [Link Incidents](#), edit the incident, add a child incident, add tasks, notes, and so on. For more information, see [incident actions](#).

If you want to customize incident layouts, see [Customize Incident View Layouts](#).

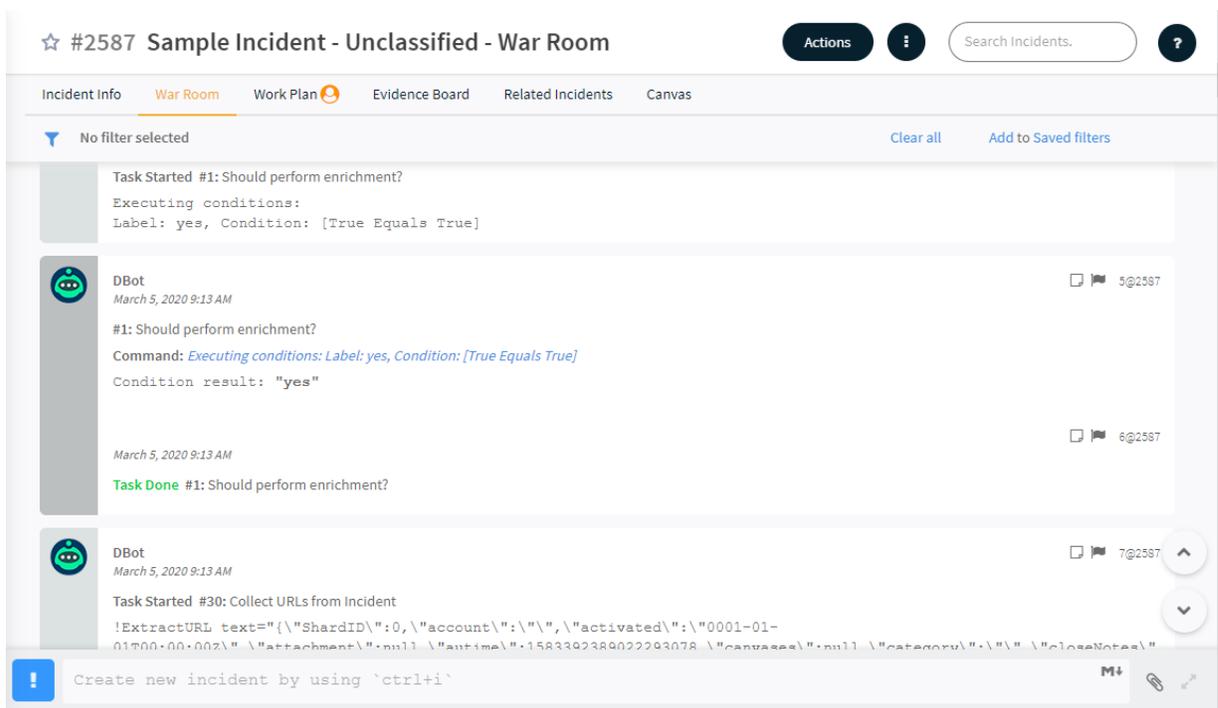
## War Room Overview

Within Cortex XSOAR, real-time investigation is facilitated through the War Room, which is powered by ChatOps and helps analysts to do the following:

- Run real-time security actions through the CLI, without switching consoles.
- Run security playbooks, scripts and commands.
- Collaborate and execute remote actions across integrated products.
- Capture incident context from different sources.
- Document all actions in one source.
- Converse with others for joint investigations.

Cortex XSOAR also provides machine learning insights to suggest the most effective analysts and command-sets. Each incident has a unique War Room.

When you open the War Room, you can see an number of entries such as commands, notes, evidence, tasks, etc, in several formats such as Markdown, HTML and so on. When Markdown, HTML or geographical information is received the content is displayed in the relevant format.



You can do the following actions for each artifact entry.

| Action                   | Description                                                                                                                                                                                                                                                                                              |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Edit                     | You can edit, format or delete your own entries. If an entry has been changed, a History link will appear where you can view all changes to the entry.                                                                                                                                                   |
| Mark as Evidence         | Opens the Mark as evidence window where you specify the evidence details to be saved in the <a href="#">Evidence Board</a> . The Evidence Board stores key artifacts for current and future analysis. You can add evidence in <b>Case Info</b> tab, the <b>Evidence Board</b> , or the <b>War Room</b> . |
| Mark as note             | Marks the incident as note. Notes can help the analyst understand why certain action was taken and assists future decisions. You can add them also in the <b>Case Info</b> tab.                                                                                                                          |
| View artifact in new tab | Opens a new tab for the artifact.                                                                                                                                                                                                                                                                        |
| Detach from task         | Removes a task from the artifact.                                                                                                                                                                                                                                                                        |
| Attach to a task         | Adds a task to the artifact.                                                                                                                                                                                                                                                                             |
| Download artifact        | Downloads an artifact according to the entry type, such txt files for text, json for a JSON entry, etc.                                                                                                                                                                                                  |
| Add tags                 | Add any relevant tags to use, which helps you find relevant information.                                                                                                                                                                                                                                 |

You can run various commands in the CLI, by typing the following:

- **!**: includes adding evidence, assigning an analyst, etc.
- **/**: includes adding notes, close an investigation, etc.
- **@**: send notifications to administrators, teams, analysts, etc.

You can edit incidents, create a report, add child incidents, and so on, as described in [Incident Actions](#).

### Filter Entities

You can filter entries by clicking . You can add any filter by selecting the checkbox or click  to remove that action. The filter menu contains three types of War Room entities by which you can filter:

- Actions
- Tags
- From

Use the And/Or toggles between the Actions, Tags and From sections.

- **And**: Use to combine two or more filters.
- **Or**: When one item is found it shows relevant entries.

You can save the filter by clicking **Add**. You can also retrieve **Saved filters**.

 *Cortex XSOAR does not index notes, chats, and pinned as evidence entries. If you want to index these entries, see [Index War Room Entries](#).*

---

## Index War Room Entries

By default, Cortex XSOAR does not index notes, chats, and pinned as evidence entries from incident War Rooms and it is not possible to find these entries in the Search Incidents bar. Use this procedure to index these entries, which also re-indexes incidents for selected months.



*Depending on the number of cases in your system and server hardware, the re-indexing operation can take a significant amount of time, during which the Cortex XSOAR server is inaccessible. It is recommended to undertake this procedure when it has a minimal impact on your organization. After completion, you should review your Cortex XSOAR server, as it may have some impact on performance.*

**STEP 1** | Log in to your Cortex XSOAR server as root or an account with sudo privileges.

**STEP 2** | Stop the Cortex XSOAR service by running the following command:

```
systemctl stop demisto
```

**STEP 3** | Make a backup copy of your demisto.conf file by running the following command:

```
cp /etc/demisto.conf /etc/demisto.conf.bak
```

**STEP 4** | Edit the `/etc/demisto.conf` file for all databases by adding the entries (highlighted in bold) in the following format:

```
"server.entries.restore": true,
 "db.index.entry.disable": false,
 "DB" : {
 "IndexEntryContent": true
 }
 "granular": {
 "index": {
 "entries": 7
 }
 }
}
```

The `granular.index.entries` total value is 7, which is split as to:

1: notes

2: chats

4: pinned as evidence

You can choose one of the values separately, or add them together for all values. For example, 7 is the total of 1 (notes) + 2 (chats) + 4 (pinned as evidence).

**STEP 5** | Save the file.

It is recommended that you validate JSON changes before committing them.

**STEP 6** | Delete the relevant War Room entries index on all databases by running the following command on each database machine:

```
rm -rf /var/lib/demisto/data/demistoidx/entries_YYYYYY
```

For example, to re-index March 2020, run the following command:

---

```
sudo -u demisto -g demisto -- /usr/local/demisto/server -stdout -restore-
index-name=entries_032020
```

To add indexing for additional months, run the same command for each month, but change the date in the command, after "entries\_". Adding months may cause re-indexing to take longer depending on the number of cases in the system.

**STEP 7 |** Start Cortex XSOAR from the command line by running one or more of the following commands:

- For the current month:

```
sudo -u demisto -g demisto -- /usr/local/demisto/server -stdout -
restore-index-name=entries_MMYYY
```

For example, to re-index March 2020, run the following command:

```
sudo -u demisto -g demisto -- /usr/local/demisto/server -stdout -restore-
index-name=entries_032020
```

- For multiple months, add the dates as CSV values:

```
sudo -u demisto -g demisto -- /usr/local/demisto/server -stdout -restore-
index-name=entries_MMYYYY,entries_MMYYYY,entries_MMYYYY
```

For example, to re-index January, February, March 2020, run the following command:

```
sudo -u demisto -g demisto -- /usr/local/demisto/server -stdout -restore-
index-name=entries_032020,entries_022020,entries_012020
```

A number of entries related to indexing appear, similar to below:

```
2019-03-21 19:00:45.651 info DB restoring 419 keys into index entries from
investigations-264/ (source: /home/circleci/.go_workspace/src/
github.com/demisto/server/repo/complexRepo/repo.go:1330)

2019-03-21 19:00:45.6649 info entry DB put in batch 78 index entries
from
investigations-264/ (source: /home/circleci/.go_workspace/src/
github.com/demisto/server/repo/complexRepo/repo.go:1363)

2019-03-21 19:00:46.4385 info entry DB put in batch 100 index entries
from
investigations-264/ (source: /home/circleci/.go_workspace/src/
github.com/demisto/server/repo/complexRepo/repo.go:1363)

2019-03-21 19:00:47.0948 info entry DB put in batch 100 index entries
from
investigations-264/ (source: /home/circleci/.go_workspace/src/
github.com/demisto/server/repo/complexRepo/repo.go:1363)

2019-03-21 19:00:47.8588 info entry DB put in batch 100 index entries
from
investigations-264/ (source: /home/circleci/.go_workspace/src/
github.com/demisto/server/repo/complexRepo/repo.go:1363)

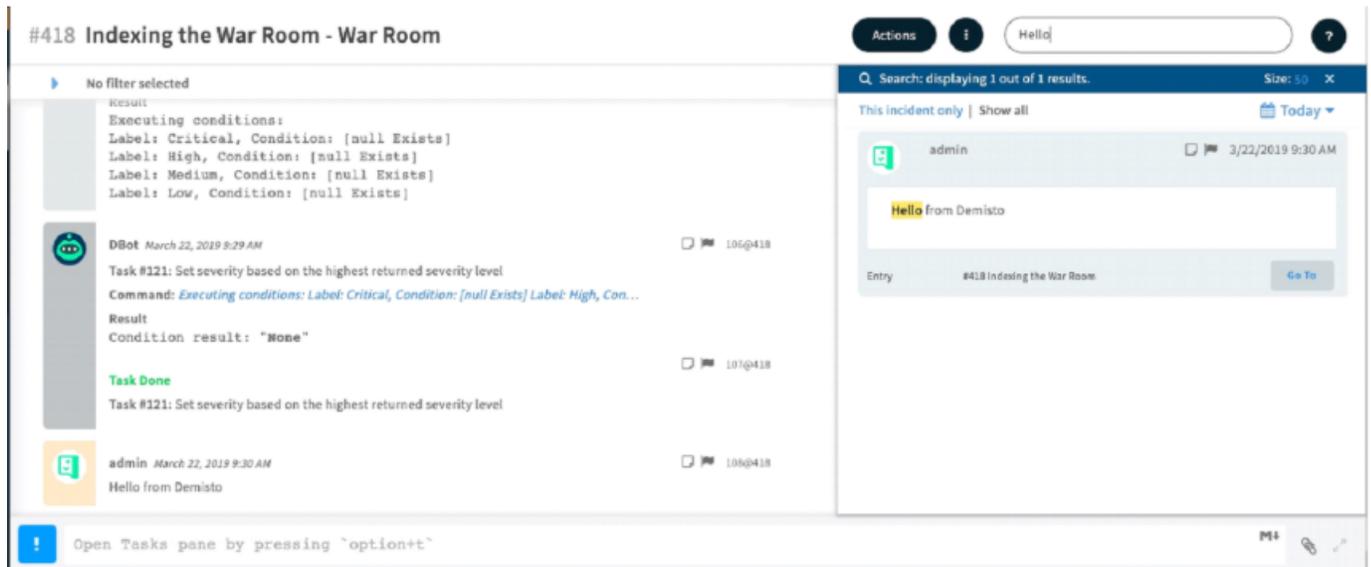
2019-03-21 19:00:48.6046 info entry DB put in batch 41 index entries
from
investigations-264/ (source: /home/circleci/.go_workspace/src/
github.com/demisto/server/repo/complexRepo/repo.go:1363)

2019-03-21 19:00:48.6047 info DB restore into [entries]
[investigations-264]
```

```
[] completed (source: /home/circleci/.go_workspace/src/github.com/demisto/server/repo/complexRepo/repo.go:1371)
```

When the re-indexing has completed, the above console messages cease and Demisto runs automatically.

**STEP 8** | Confirm that you can search your case comments through the search bar.



**STEP 9** | Stop the service by using CTRL-C as the Cortex XSOAR server is running locally from the command line.

**STEP 10** | Start Cortex XSOAR by running the following command:

```
systemctl start demisto
```

## Add a Custom Widget in the War Room

You can add a custom widget in the War Room using an automation script. After creating the script in the **Automation** page, to add a custom script based widget, you need to run a command in the War Room.

**STEP 1** | Create a Custom Widget Using an Automation Script.

**STEP 2** | Go to the **War Room** and run the following command:

```
!<scriptName>
```

where scriptName is the name of the automation script you created in step 1.

In the following example, you need to add a custom widget that shows you the severity of the indicators in an incident, as a bar chart.

Use the following script:

```
commonfields:
id: ee3b9604-324b-4ab5-8164-15ddf6e428ab
version: 49
```

```

name: IndicatorWidgetBar
script: |-
 # Constants
 HIGH = 3
 SUSPICIOUS = 2
 LOW = 1
 NONE = 0

 indicators = []
 scores = {HIGH: 0, SUSPICIOUS: 0, LOW: 0, NONE: 0}
 incident_id = demisto.incidents()[0].get('id')

 foundIndicators = demisto.executeCommand("findIndicators",
{"query": 'investigationIDs:{}'.format(incident_id), 'size':999999})[0]
['Contents']

 for indicator in foundIndicators:
 scores[indicator['score']] += 1

data = {
 "Type": 17,
 "ContentsFormat": "bar",
 "Contents": {
 "stats": [
 {
 "data": [
 scores[HIGH]
],
 "groups": None,
 "name": "high",
 "label": "incident.severity.high",
 "color": "rgb(255, 23, 68)"
 },
 {
 "data": [
 scores[SUSPICIOUS]
],
 "groups": None,
 "name": "medium",
 "label": "incident.severity.medium",
 "color": "rgb(255, 144, 0)"
 },
 {
 "data": [
 scores[LOW]
],
 "groups": None,
 "name": "low",
 "label": "incident.severity.low",
 "color": "rgb(0, 205, 51)"
 },
 {
 "data": [
 scores[NONE]
],
 "groups": None,
 "name": "unknown",
 "label": "incident.severity.unknown",
 "color": "rgb(197, 197, 197)"
 }
],
 "params": {

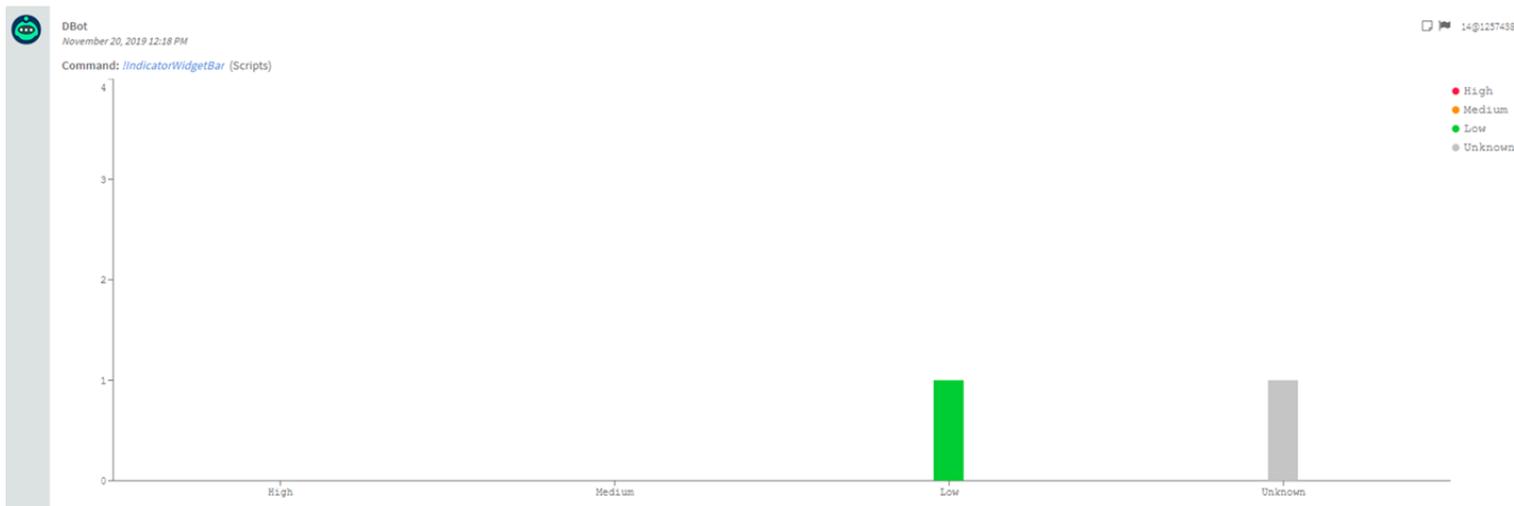
```

```
 "layout": "horizontal"
 }
}

demisto.results(data)
type: python
tags:
- dynamic-section
enabled: true
scripttarget: 0
subtype: python3
runonce: false
dockerimage: demisto/python3:3.7.3.286
runas: DBotWeakRole
```

Create a new automation by adding the script and then in the War Room run the `!IndicatorWidgetBar` command.

The custom widget appears in the War Room.



## Work Plan

The Work Plan is visual representation of the running [Playbook](#) that is assigned to the incident. Playbooks enable you to automate many of your security processes, including, but not limited to handling your investigations and managing your tickets. Work Plans enable you to monitor and manage a Playbook work flow, and add new tasks to tailor the Playbook to a specific investigation.

When clicking the **Follow** checkbox you can see the Playbook executing in real-time.

In the Work Plan you can do the following:

- View [inputs and outputs](#) of a playbook.
- View, create, and edit a [playbook task](#) for each required step.

When you create a task, add a name, automation, and description. The name and description should be meaningful so that the task corresponds to the data that you are collecting.

For each task you can do the following:

Designate tasks as complete either manually or by running a script.

Assign an owner for a task.

Set a due date for the task.

Add comments and completed notes, as required.

- Re-run the playbook, zoom in and out, and export to a PNG format.

## Link Incidents

You can link or unlink incidents through the following:

- In the [Related Incidents](#) tab.
- A [pre-process rule](#), so that as soon as an incident is ingested into Cortex XSOAR you can link incidents.
- Using the [CLI](#).

After you link the incident, you can view linked incidents in the **Case Info** tab.

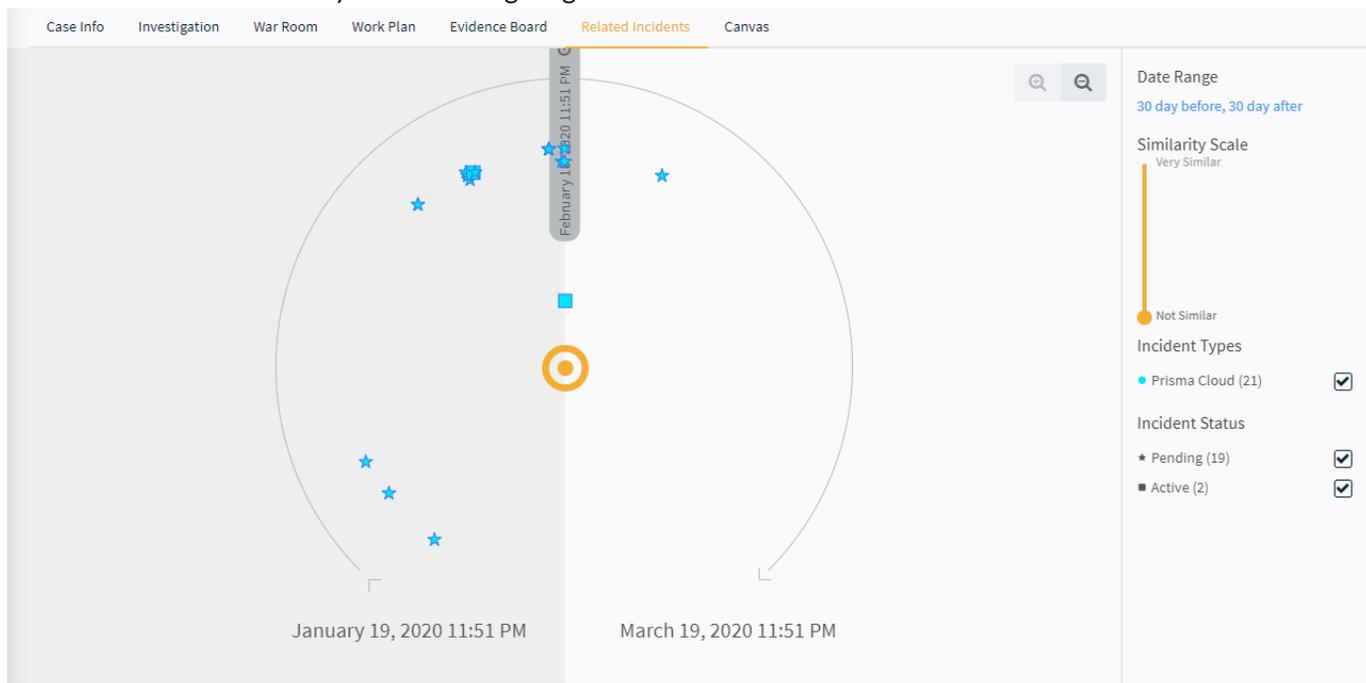
| Linked Incidents (1) |                            |       |   |
|----------------------|----------------------------|-------|---|
| ID                   | Name                       | Owner | T |
| #91                  | Sample Incident - Phishing |       | P |

## Manage Related Incidents

Related incidents are a visual representation of incidents that share similar characteristics, such as malicious indicators, or part of a single phishing campaign. Viewing related incidents in a single view enables you to consolidate the investigation by deduplicating and linking related incidents to the incident you are viewing. Linking incidents helps you assess whether the action taken is effective.

### Using the Related Incidents Map

Go to the incident that you are investigating and click **Related Incidents**.



## Understanding the Related Incidents Map

- The incident you are currently investigating is at the center of the Related Incidents map, surrounded by the related incidents. The more similar a related incident, the closer it is to the center.
- The incidents are categorized according to incident status (pending, active, and closed) and type (such as malware, phishing, and so on). In this example, phishing is categorized:

| Shape                                                                             | Status         |
|-----------------------------------------------------------------------------------|----------------|
|  | Pending status |
|  | Active status  |
|  | Closed status  |

- The map has a time spectrum. Incidents on the right side of the map are newer than the current incident, and the incidents on the left are older. Related incidents are spread across the spectrum according to the time the incident was created. The time scope is 30 days before and 30 days after the currently investigated incident. You can modify the range by using the **Date Range**.
- Use the **Similarity Scale** to display related incidents that are more similar or less similar to the current incident.
- Hover over a related incident to view detailed information.
- Click an incident to view a comparison of the two incidents, which shows instances of similar indicators between the incidents. You can click multiple incidents by using **ctrl + click** or **command + click**. In the Similarities window, you can pair as **Linked** or as **Duplicate**. The incident appears as linked in the Linked Incidents table in the **Case info** tab.

If you want to build your own related incidents and indicators a layout of your choice, use the [Canvas](#). The **Related Incidents** page is orientated towards exploration and searching for similar data.

You can configure an allow list or an ignore list for which incident fields to use for related incidents, as described in [Configure Incident Fields for Related Incidents](#).

### Configure Incident Fields for Related Incidents

You can configure an allow list or an ignore list for which incident fields to use for related incidents. If you define an allow list, related incidents only use specified fields for calculation. If you define an ignore list, related incidents are calculated without the specified fields.

**STEP 1** | Select **Settings > About > Troubleshooting > Add Server Configuration**.

**STEP 2** | Add the following keys and values:

| List type   | Key                           | Value                                                      |
|-------------|-------------------------------|------------------------------------------------------------|
| Allow list  | incident.metadata.whitelist   | A comma-separated list of fields to add to the allow list. |
| Ignore list | incident.metadata.ignore.list | A comma-separated list to exclude from related incidents.  |

## Link and Unlink incidents in the CLI

You can link and unlink incidents in the CLI. The linked incidents appear in the **Case Info** tab.

**STEP 1** | To link an incident, in the CLI, type the following command:

```
!linkIncidents linkedIncidentIDs=<id> action=link
```

Confirmation of the linked incident appears in the War Room and in the **Case Info** tab. In this example, we have linked the incident ID#3



DBot  
March 17, 2020 11:58 AM

Command: `!linkIncidents linkedIncidentIDs="3" action="link"` (Builtin)  
1 incidents linked

Linked Incidents (1)

| ID | Name                                         | Owner |
|----|----------------------------------------------|-------|
| #3 | Live friend trouble take billion many shake. |       |

**STEP 2** | To unlink the incident, type the following command:

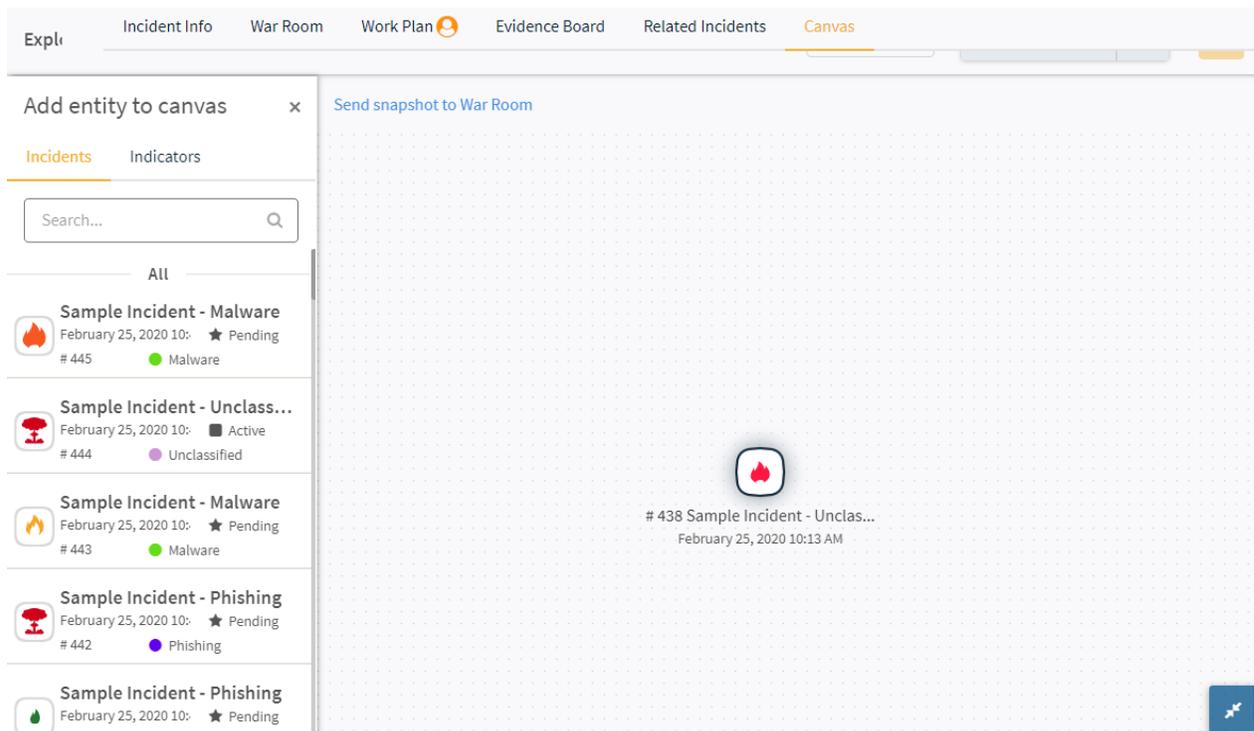
```
!linkIncidents linkedIncidentIDs=<id> action=unlink
```

## Investigate Using the Canvas

The Investigation Canvas enables you to visually map an incident, its elements, and progression path, combining analyst intelligence with machine learning.

To access the investigation canvas, click **Canvas** from the incident you want to investigate. The incident appears on the canvas display. In the **Add entity to canvas** section, DBot provides suggested indicators and incidents that might be connected or relevant to the current incident for you to add to the canvas.

Searches are server side searches.



Explo | Incident Info | War Room | Work Plan | Evidence Board | Related Incidents | **Canvas**

Add entity to canvas x Send snapshot to War Room

Incidents Indicators

Search...

All

- Sample Incident - Malware  
February 25, 2020 10: Pending  
# 445 Malware
- Sample Incident - Unclas...  
February 25, 2020 10: Active  
# 444 Unclassified
- Sample Incident - Malware  
February 25, 2020 10: Pending  
# 443 Malware
- Sample Incident - Phishing  
February 25, 2020 10: Pending  
# 442 Phishing
- Sample Incident - Phishing  
February 25, 2020 10: Pending

# 438 Sample Incident - Unclas...  
February 25, 2020 10:13 AM

---

## Incident Suggestions

The incidents are calculated according to the related incidents algorithm, which are based on several factors:

- Common labels
- Common indicators
- Common incident custom fields

You can add the incidents by dragging and dropping the incident onto the canvas.

## Indicator Suggestions

The indicators are determined according to the following factors (in this order):

1. Indicators with a bad reputation from the original incident (the incident that initiated the investigation).
2. Indicators that are shared between incidents that you added to the canvas.
3. The malicious ratio, which is the ratio between the indicators that appear in incidents with a bad reputation, compared to the total number of incidents in Cortex XSOAR.

You can add the indicators by dragging and dropping the indicators onto the canvas.

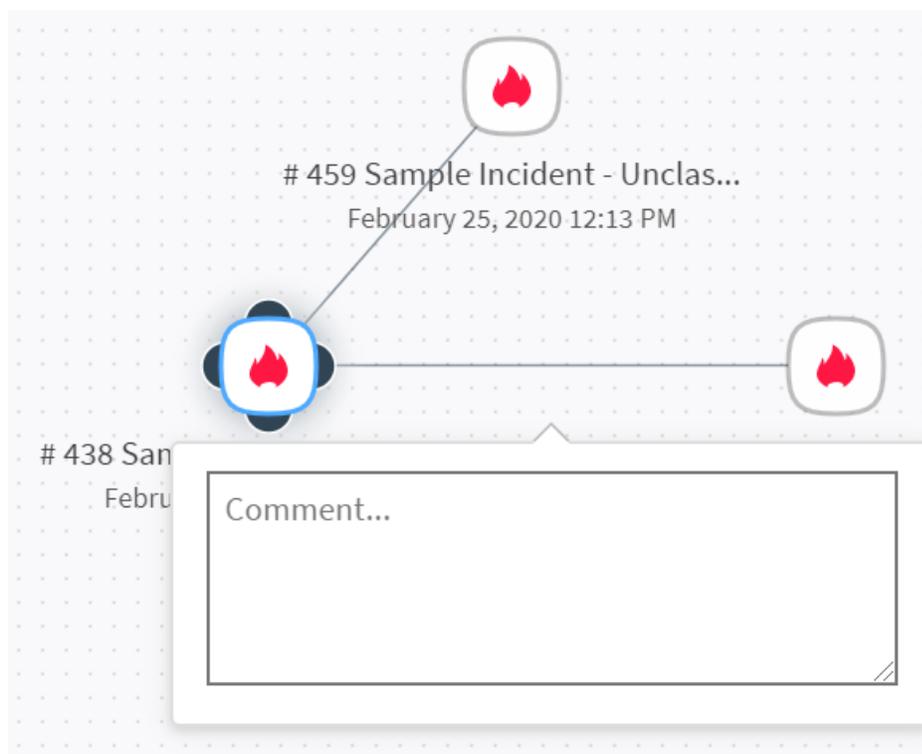


*You can [Edit Dbot Incident and Indicator Suggestions](#) in the Entity Library.*

## Key Features

You can do the following:

- **Auto Populate the Canvas** with related incidents, suspicious URLs and so on by using machine learning.  
The closer an entity appears to the center, the more closely related it is to the investigated incident.
- **View an incident and indicator:** view details of incident and indicator, including various actions in the [Dbot Suggestions: Quick View Window](#).
- **Connect incidents:** connect each incident by linking each incident and use comments on entity connections to communicate important information with team members by adding notes to connectors between entities.



- **Dynamic Connections:** when you rearrange entities on the canvas, the connections dynamically move with the entities. Connections that are dotted lines indicate that the indicator is part of the investigation, or two incidents are defined as related incidents. These connections are dynamic, which means if one entity is an IP address and you add that IP address to the allow list after it was added to the canvas, the dotted-lined connection is automatically removed.
- **Capture the Canvas as an image:** capture and study the incident by clicking **Export to PNG**.

## Auto Populate the Canvas

Cortex XSOAR can automatically populate the Canvas with related entities using machine learning. If your canvas is already populated, auto populating it deletes all of the existing content.

**STEP 1** | Go to the **Canvas** tab of the incident you are investigating and click **Auto populate**.

**STEP 2** | If you want to customize the canvas, click **Customize** and select the following:

- If, and how many, related incidents appear.
- The maximum distance over which items are included in the canvas in the **Similarity Max Distance** field.  
By default the distance is set to 0.8. The closer the score is to 1, the less related they are to the incident.
- **Linked incidents**
- **Bad and suspicious common indicators**
- Configure the threshold above which an indicator is ignored in the **Indicators Ignore Threshold**.

**STEP 3** | Click **Auto populate**.

The closer an entity appears to the center, the more closely related it is to the investigated incident.

## Dbot Suggestions: Quick View Window

The Quick View window displays information for the entity selected on the canvas, either an incident or an indicator, each of which have DBot suggested indicators.

You can highlight entities on the canvas to show visually how the incident progressed.

Searches performed in the Quick View pane are client-side searches.

The screenshot displays the Cortex XSOAR interface. At the top, there are navigation tabs: Case info, Investigation, War Room, Work Plan, Evidence Board, Related Incidents, and Canvas. The 'Canvas' tab is active. Below the navigation, there's a search bar and an 'Auto populate' button. The main area is divided into three panels. On the left, there's a panel titled 'Add entity to canvas' with a search bar and a list of incidents. The middle panel shows a grid of entities on the canvas. On the right, the 'Quick View' window is open, showing details for a selected incident. The 'Quick View' window has a 'Details' tab and an 'Indicators (10)' tab. Under the 'Indicators (10)' tab, there's a search bar and a list of indicators. The indicators are categorized into 'Domain (0/4)', 'Email (0/2)', and 'File (0/1)'. The domain indicators listed are: nodomain.net, kloshpro.com, www.kloshpro.com, and demisto.int, each with a 'Last Seen' date and time and a checkmark icon.

### Incident Quick View

You can view basic information, such as type, severity, time line information labels, and indicators. The indicators that DBot suggests to add to the canvas for this incident are determined according to the following factors (in this order):

1. Indicators with a bad reputation from the current (selected) incident.
2. The malicious ratio, which is the ratio between the indicators that appear in incidents with a bad reputation, compared to the total number of incidents in Cortex XSOAR.

### Indicator Quick View

You can view source information, hashes, known history, comments and do certain actions such as run scripts, delete, exclude and so on.

The indicators that DBot suggests to add to the canvas for the selected indicator are determined according to the following factors (according to this order):

1. Relations between all canvas investigation contexts. For example, if a hostname and IP address are associated with the same endpoint, the context key is suggested as an indicator.
2. An ssdeep with 50% or higher similarity.

You can [Edit Dbot Incident and Indicator Suggestions](#) in the **Quick View** window.

## Edit Dbot Incident and Indicator Suggestions

In the Investigation Canvas, Dbot provides suggestions in the Entity Library section and the **Quick View** window. You can change these suggestions by adding keys and values when adding a server configuration.

**STEP 1** | Select **Settings > About > Troubleshooting > Add Server Configuration**.

**STEP 2** | Add the following keys and values as required:

| Key                                                       | Description                                                                                                                   |
|-----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| canvas.suggestions.IncidentIndicatorSuggestions.max       | Maximum number of suggestions for malicious suggestions. Default: 5<br><br>Relevant for Library and Quick View.               |
| canvas.suggestions.IncidentMutualIndicatorSuggestions.max | Maximum number of suggestions for common indicators between incidents. Default: 5<br><br>Relevant for Library and Quick View. |
| canvas.suggestions.IndicatorIndicatorSuggestions.max      | Maximum number of suggestions for indicators. Default: 5<br><br>Relevant for Quick View.                                      |
| ml.suggestions.canvas.leftpane.incidents.limit            | Maximum number of incidents in the Quick View window. Default: 10<br><br>Relevant for Quick View.                             |

## Incident Actions

In an incident, you can undertake a number of actions, such as edit the incident, add a child incident, add tasks, notes, and so on.

When clicking **Actions** you can undertake the following actions:

| Action             | Description                                                                                                                                                                                                                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Edit               | Edit the incident as required.                                                                                                                                                                                                                                                                       |
| Report             | Create a report to capture investigation specific data and share it with team members.                                                                                                                                                                                                               |
| Add child incident | Adds a child incident to the incident.<br><br>Child investigations are used to compartmentalize sensitive War Room activity. You can create child investigations to collaborate discreetly with a select group of people on a specific topic of investigation. Child investigations are also used in |

| Action                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                   | <p>situations where a secondary investigation is needed and its content may add too much "noise" in the original investigation.</p> <p>You can also create child investigations from the CLI using the / <b>investigation_child_create</b> command.</p> <p>To turn the child investigation to a discrete investigation, select the <b>Restricted</b> checkbox.</p> <p> <i>Closing a parent investigation also closes all associated child investigations.</i></p> |
| <a href="#">Restrict incident</a> | Restrict an investigation to the incident owner and team.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Close incident                    | Marks the incident as closed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Delete                            | Deletes the incident                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

When clicking  you can undertake the following actions:

| Action                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Quick View                     | You can see a summary of the incident, timeline information, labels, and indicators.                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="#">Incident Tasks</a> | Add tasks for users to complete as part of an investigation.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Systems                        | Details of any D2 Agents that are deployed to perform forensic tasks on machines.                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Team                           | Add team members to the incident.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Context Data                   | <p>View context data. The context is a map (dictionary) that is created for each incident and is used to store structured results from the integration commands and automation scripts. The context keys are strings and the values can be strings, numbers, objects, and arrays.</p> <p>You can use context data to:</p> <ul style="list-style-type: none"> <li>• Pass data between playbook tasks.</li> <li>• Capture the important structured data from automations and display the data in the incident summary</li> </ul> |

You can also edit or add actions in the **Case Info/Incident Info** field.

## Evidence Handling

You can view or designate any entity as evidence which enables you to reconstruct attack chains and piece together key pieces of verification for root cause discovery.

In the War Room you can mark any entity as evidence by clicking the flag next to each entry. You can view the evidence in the War Room or open the evidence entry from the Evidence Board. When adding evidence you need to add a description which should contain enough details that can be used for future reference.

---

Adding a tag helps you to find the evidence by searching for the tag. You can also add an occurrence date and time.

### The Evidence Board

The Evidence board stores key artifacts for current and future analysis. You can view and manage evidence entities that were detected in the War Room and designated as Evidence.

You can search for evidence and select the date range when the evidence occurred.

Evidence can be viewed in **Table View** or **Summary View**. In the **Table View**, you can remove, export, or show in the War Room. In the **Summary View** you can remove or edit the evidence.

## Incident Tasks

Incident tasks are tasks for users to complete as part of an investigation, which are split according to the following:

- **Playbook Tasks:** you can view, assign an owner, complete, and set a due date for playbook tasks that require attention.
- **To Do Tasks:** create tasks for users to complete as part of an investigation, and which are not attached to the incident's playbook. A playbook can finish running and an incident can be closed even if the incident contains open To-Do tasks.

You can [Create a To-Do Task](#) directly from the incident Case (incident) info tab or in the To Do Task section.

Alternatively, you can create To Do tasks in the command line.

### Create a To-Do Task

You can create To-Do tasks directly in the incident from either the Case (Incident) Info tab or the Incident Tasks window. You can also create To Do Tasks using the command line.

**STEP 1** | From the incident, click  and then click **Incident Task**.

**STEP 2** | In the Incident Tasks window, click the **To Do Tasks** tab.

You can also add To Do Tasks from the Case (Incident) Info tab if you have [customized](#) the incident to include To Do tasks.

**STEP 3** | Click **Add a task**.

**STEP 4** | Add the **Task Details** as required:

| Parameter        | Description                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Task Name        | A meaningful name for the task (mandatory).                                                                                                 |
| Task Description | A meaningful description for the task that provides sufficient information for the assignee to complete the task.                           |
| Assignee         | The user to assign to the task. You can only assign a single user per task.                                                                 |
| Set due date     | The due date for the task. If the task is not completed by this date, it is marked as overdue but is not a roadblock for the investigation. |

| Parameter           | Description                      |
|---------------------|----------------------------------|
| Tag the result with | Tags to apply to the To-Do task. |

**Task Details** ✕

**Task Name\***

**Task Description**

**Assignee**

Admin ▾

**Set due date**

February 26, 2020, 4:24 PM

**Tag the result with**

Phishing ✕

▾

Cancel
Save

**STEP 5 | Click Save.**

If you have [customized](#) the incident, you can see the **To Do Tasks** from the Case (Incident) Info tab.

## Incident Fields

Use Incident Fields to accept or populate incident data coming from incidents. You create fields for information that arrives from third party integrations in which you want to insert information. The fields are added to Incident Type layouts and are mapped using the Classification and Mapping feature.

Incident Fields can be populated by the incident team members during an investigation, at the beginning of the investigation, or prior to closing the investigation.

*Creating Incident Fields is an iterative process in which you continue to create fields as you gain a better understanding of your needs and the information available in the third party integrations that you use.*

You can set and update all system incident fields using the `setIncident` command, of which each field is a command argument.

### Incident Field Types

You can add the following field types, when adding a new field.

- Attachments : enables adding an attachment, such as .doc, malicious files, reports, images of an incident, etc.
- Boolean (checkbox)
- Date picker
- Grid (table): include an interactive, editable grid as a field type for selected incident types or all incident types.
- HTML
- Long text
- Markdown
- Multi select
- Number: can contain any number. The default number is 0. Any quantity can be used.
- Role: roles assigned to the incident determine which users (by the role to which they are assigned) can view the incident.
- Short text: maximum of 60,000 characters.
- Single select
- Tags
- Timer/SLA: view how much time is left before an SLA becomes past due, as well as configure actions to take in the event that the SLA does pass.
- URL
- User : a user in the system to state a manager or fallback.

## Basic Settings

The following table lists the fields that appear in the Basic Settings page, and their descriptions. The Basic Settings page is available for the following field types:

- Long text
- Mult select
- Short text
- Single select
- Tags

| Name        | Description                                                           |
|-------------|-----------------------------------------------------------------------|
| Placeholder | Define the text that appears in the field before users enter a value. |
| Values      | A comma separated list of values that are valid values for the field. |

## Timer/SLA Fields

The following table lists the fields specific to Timer/SLA fields, and their descriptions.

| Name           | Description                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SLA            | Determine the amount of time in which this item needs to be resolved. If no value is entered, the field serves as a counter.                                                     |
| Risk Threshold | Determine the point in time at which an item is considered at risk of not meeting the SLA. By default, the threshold is 3 days, which is defined in the global system parameter. |

| Name              | Description                                                                                                                                                                                                                                                                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Run on SLA Breach | <p>In the Run on SLA Breach field, select the script to run when the SLA time has passed. For example, email the supervisor or change the assignee.</p> <p> <i>Only scripts to which you have added the SLA tag appear in list of scripts that you can select.</i></p> |

## Attributes Parameters for Incident Fields

The following tables list the fields that are common to all Incident Fields.

| Name                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Script upon change             | The script that dynamically changes the field value when script conditions are met. For a script to be available, it must have the <b>field-change-triggered</b> tag, when defining an automation. For more information, see <a href="#">Field Trigger Scripts</a> .                                                                                                                                                                                                                                                                |
| Field display script           | Determines which fields display in forms, as well as the values that are available for single-select and multi-select fields. For more information, see <a href="#">Create Dynamic Fields in Incident Forms</a> .                                                                                                                                                                                                                                                                                                                   |
| Add to incident types          | Determines for which incident types this field is available. By default, fields are available to all incident types. To change this, clear the <b>Associate to all</b> checkbox and select the specific incident types to which the field is available.                                                                                                                                                                                                                                                                             |
| Default display on             | Determines at which point the field is available. For more information, see <a href="#">Incident Field Examples</a> .                                                                                                                                                                                                                                                                                                                                                                                                               |
| Edit Permissions               | Determines whether only the owner of the incident can edit this field.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Make data available for search | <p>Determines if the values in these fields are available when searching.</p> <p> <i>In most cases, Cortex XSOAR recommends that you select this checkbox so values in the field are available for indexing and querying. However, in some cases, to avoid adverse affects on performance, you should clear this checkbox. For example, if you are ingesting an email to an email body field, we recommend that you not index the field.</i></p> |
| Add as optional graph          | Determine if you can create a graph based on the contents of this field. This field does not appear for all field types.                                                                                                                                                                                                                                                                                                                                                                                                            |

## Incident Field Examples

The following section shows several examples of common fields that are used in real-life incidents.

### False Positive

Below is an example of a mandatory Incident field "False Positive" to be filled at time of Incident Close. The Field can have a value YES or NO and the SOC admin should be able to query or run report based on this field. After this field is added, all incidents will need to have this filled in before an incident can be marked closed.

## SLA Fields

The following SLA field can be used to trigger a notification when the status effecting the SLA of an incident changes. If the SLA is breached, we have configured the field such that an email is sent to the owner's supervisor.

## Troubleshooting Conflicts with Custom Incident Fields

When trying to download a content update, you receive the following message:

**Warning: content update has encountered some conflicts**

This occurs when a content update has an incident field with the same name as a custom incident field that already exists in Cortex XSOAR.

### Solution

Click **Install Content** to force the update and retain your custom incident field. The content update will install without the system version of the incident field.

## Create a Custom Incident Field

You can define custom incident fields based on the information you want to display in your Incident Type layouts, as well as the information ingested from third party integrations.

**STEP 1** | Select **Settings > ADVANCED > Fields > + New Field**.

Depending on the field type, you can determine if the field contents are case-sensitive, as well as if the field is mandatory.

**STEP 2** | In the **Field Type** field, from the dropdown list select the [Incident Field Types](#).

**STEP 3** | Complete the following parameters:

| Field      | Description                                                                          |
|------------|--------------------------------------------------------------------------------------|
| Field Name | A descriptive name indicating the information that the field contains.               |
| Tooltip    | (Optional) Additional information you want to make available to users of this field. |

**STEP 4** | If relevant to the field type, add the [Basic Settings](#).

If adding a grid, see [Create a Grid Field for an Incident Type](#).

**STEP 5** | In the **Attributes** tab, add the [attribute parameters](#).

**STEP 6** | Click **Save**.

**STEP 7** | To add the field to the incident, go to **Settings > ADVANCED > Incident Types**, select the incident, and click **Edit Layout**.

**STEP 8** | In the Library dialog box, in the **Cortex XSOAR Sections** tab, drag and drop **+ New Section** on to the required tab.



**STEP 9** | In the **Incident** field tab, drag and drop the field that you have created into the **New Section**.

## Create a Grid Field for an Incident Type

The grid field enables you to view and edit a table when adding it to an incident. You can create a grid field and add it to an incident type.

**STEP 1** | Select **Settings > ADVANCED > Fields > + New Field**.

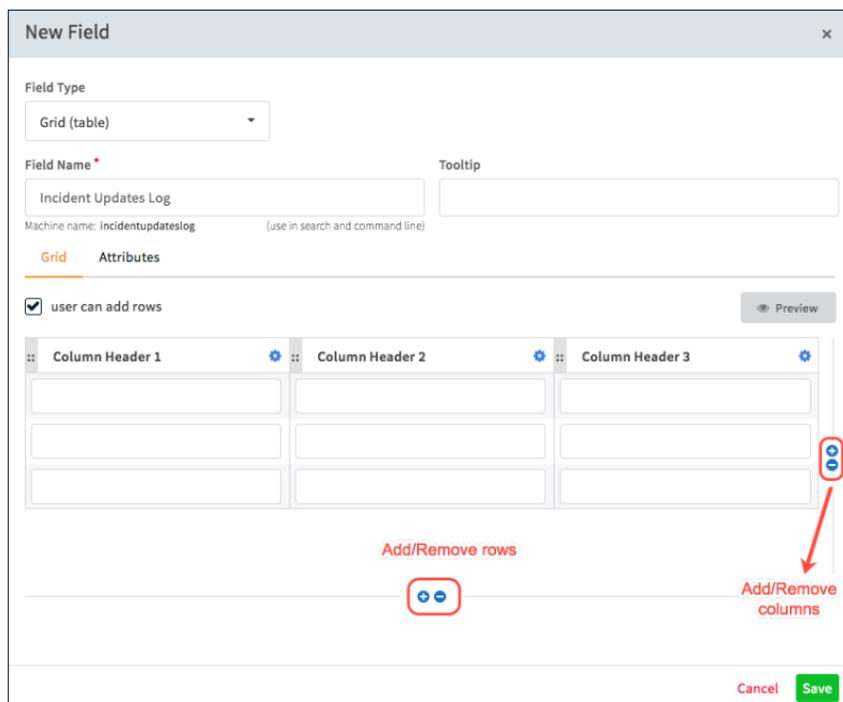
**STEP 2** | From the New Incident Field window, in the **Field Type** field, from the drop down list, select **Grid (table)**.

**STEP 3** | Complete the following parameters:

| Parameter         | Description                                                                               |
|-------------------|-------------------------------------------------------------------------------------------|
| Field Name        | A meaningful name for the grid field.                                                     |
| Tooltip           | (Optional) A brief descriptive message that explains what the field is and how to use it. |
| User can add rows | (Optional) Enables users to add/remove rows in the grid.                                  |

**STEP 4** | In the **Grid** tab, add or remove the required rows and columns.

How you design the grid determines how it appears to users. If **users can add rows**, they can add rows, but not columns.



**STEP 5** | Configure each column by selecting the required field types, such as short text, boolean, URL, etc, for each column.

For example, we want to define three mandatory columns: Name, Location, and Date. If you select the Lock checkbox, the value for that field is static (not editable). If you do not select the Lock checkbox (default), users can perform in-line editing.

×
Column Settings

**Column Name \***

Mandatory

**Field Type**

Multi select

Lock

**Values (comma separated) \***

New York, Boston, Paris, Berlin

"Select" will be displayed as default option

Cancel
Save

×
New Incident Field

**Field Type**

Grid (table)

**Field Name \*** **Tooltip**

Incident Updates Log

Machine name: incidentupdateslog (use in search and command line)

Grid   Attributes

---

user can add rows

Preview

| Name *                                   | Location * | Date *                                                       |
|------------------------------------------|------------|--------------------------------------------------------------|
| <input style="width: 90%;" type="text"/> | Select ▼   | <input style="width: 95%;" type="text" value="Choose date"/> |
| <input style="width: 90%;" type="text"/> | Select ▼   | <input style="width: 95%;" type="text" value="Choose date"/> |
| <input style="width: 90%;" type="text"/> | Select ▼   | <input style="width: 95%;" type="text" value="Choose date"/> |

+ -

Cancel
Save

**STEP 6 |** In the **Attributes** tab, add the [attribute parameters](#).

For example, we have added the **emailFieldTriggered** script and made the field available to Access, Malware, and Network incident types.

**Field Type**

Grid (table) ▾

**Field Name \*** **Tooltip**

Incidents Updates Log

Machine name: incidentsupdateslog (use in search and command line)

Grid **Attributes**

---

Script upon change emailFieldTriggered ▾ ✕ ?

Field display script Choose script ▾ ?

Add to incident types

Access ✕ Malware ✕ Network ✕ ▾

Associate to all

Default display on:

New / Edit  Close  Both

Edit Permissions

Only owner can edit

Indexing

Make data available for search

Hide

**STEP 7 |** Click **Save**.

## Use Scripts with the Grid Field

You can use scripts to manipulate and populate data in the Grid field. In this example, we will use the following scripts:

- Automatically populate a column value when the grid is changed.
- Create a new row in the grid manually or as part of a playbook.



*If you select the Lock checkbox for a column, only a script can populate the values for that column. If a column is unlocked (default), the column values can be entered manually (by users), or by a script. For a script to be available in the Script upon change drop-down menu, it must have the **field-change-triggered** tag.*

### Grid Field Script Example

In this example, the grid is a shift summary for analysts, who can add comments for the incident during their shift. We want to use a script to automatically populate the **Date Logged** column with the current date when a user adds a new row to the grid.

#### Sample script

The **shiftSummariesChange** script is called with an old value and a new value. The script operates in the following phases:

- The script gets all new rows, and sets the Date Logged field to now (current day).
- For each existing row, if the name matches, but the findings column is not updated, the Date Logged column is also updated.
- The Shift Summaries field is saved with the new values using the **setIncident** command.

```

var newField = args.new ? JSON.parse(args.new) : [];

// if line(s) added, set "datelogged" to now.
if (oldField.length < newField.length) {
 // for each new line change date.
 for(var i=oldField.length; i < newField.length; i++) {
 newField[i].datelogged = new Date().toISOString();
 }
}
var columnName = "findings";
// for each old line if the "columnName" has changed, change date to now.
for(var i=0; i < oldField.length; i++) {
 if (newField[i] && oldField[i].fullname === newField[i].fullname &&
 oldField[i][columnName] !== newField[i][columnName]) {
 newField[i].datelogged = new Date().toISOString();
 }
}
var newVal = {};
newVal[args.cliName] = newField;
executeCommand("setIncident", newVal);

```

### Add a Row to a grid Using a Script

During playbook execution if a malicious finding is discovered, you want to add that finding to a grid. You can use a script in the playbook to add a new row to the grid with the malicious finding.

#### Sample Script

This is a Python script, which requires 2 arguments:

- **fieldCliName**: the machine name for the field for which you want to add a new row.
- **Row**: the new row to add the grid. This is a JSON object in lower case characters, with no white space.

```

fieldCliName = demisto.args().get('field')
currentValue = demisto.incidents()[0]["CustomFields"][fieldCliName];

if currentValue is None:
 currentValue = [json.loads(demisto.args().get('row'))]
else:
 currentValue.append(json.loads(demisto.args().get('row')))

val = json.dumps({ fieldCliName: currentValue })
demisto.results(demisto.executeCommand("setIncident", { 'customFields':
 val })))

```

## Field Trigger Scripts

Incident fields can be associated with trigger automation scripts that can check for field change conditions and take actions based on the change. These scripts can perform any action, such as dynamically changing the field value, notifying the responder when an incident severity has been changed, etc, when the conditions are met. For example, when using the **ChangeRemediationSLAOnSevChange** automation, it changes the Remediation SLA of an incident, when the severity of the incident changes for any reason.

Automations can be created in Python, PowerShell, or JavaScript in the Automation page. To use a field trigger automation, you need to add the field-change-triggered tag when creating the automation. You can then add the automation in the [Attributes](#) tab, when you edit or [Create a Custom Incident Field](#). If you did not add the tag when creating the automation, it cannot be added.

Cortex XSOAR comes out-of-the-box with the following field change scripts in the Automation page:

- **ChangeRemediationSLAOnSevChange**
- **emailFieldTriggered**: Changes the incident field when the email body or subject changes.
- **StopTimeToAssignOnOwnerChange**: Stops the Time to Assignment SLA field, as soon as an owner was assigned to an incident.



The ExtraHop Reveal(x) Content Pack contains a field trigger script, which only tracks the incident if it is from an ExtraHop Detection.

A common use case is to create the following automation that verifies that changes made by a playbook only will take place and not by a user manually.

```
args = demisto.args()
user = args["user"]
if user:
 demisto.executeCommand("setIncident", {args["name"]: args["old"]})
```

The automation checks who made the change using the **user** field. The **name** argument returns the field name, so that it can be attached to multiple incident fields, and block changes to them, without the need to have a different automation for each field.

### Automation Arguments - Related Information

When an automation is triggered on a field, it has the following triggered field information available as arguments (args):

| Argument               | Description                                                                                 |
|------------------------|---------------------------------------------------------------------------------------------|
| <b>associatedToAll</b> | If the field is associated to all or some incidents. Value: <b>true</b> or <b>false</b> .   |
| <b>associatedTypes</b> | An array of the incident types, with which the field is associated.                         |
| <b>cliName</b>         | The name of the field when called from the command line.                                    |
| <b>description</b>     | The description of the field.                                                               |
| <b>isReadOnly</b>      | Specifies whether the field is non editable. Value: <b>true</b> or <b>false</b> .           |
| <b>name</b>            | The name of the field.                                                                      |
| <b>new</b>             | The new value of the field.                                                                 |
| <b>old</b>             | The old value of the field.                                                                 |
| <b>ownerOnly</b>       | Specifies that only the creator of the field can edit. Value: <b>true</b> or <b>false</b> . |
| <b>placeholder</b>     | The placeholder text.                                                                       |
| <b>required</b>        | Specifies whether this is a mandatory field. Value: <b>true</b> or <b>false</b> .           |
| <b>selectValues</b>    | If this is a multi select type field, these are the values the field can take.              |
| <b>system</b>          | Whether it is a Cortex XSOAR defined field.                                                 |

| Argument                     | Description                                                                       |
|------------------------------|-----------------------------------------------------------------------------------|
| <code>type</code>            | The field type.                                                                   |
| <code>unmapped</code>        | Whether it is not mapped to any incident.                                         |
| <code>useAsKpi</code>        | Whether it is being used for tracking KPI on an incident page.                    |
| <code>username</code>        | The username of the user who triggered the script.                                |
| <code>validationRegex</code> | Whether there is a regex associated for validation the values the field can hold. |

### Script Limitations

- Trigger scripts cannot close incidents.
- Post-processing scripts can modify an incident, but if a modified field has a trigger script, it is not called.
- Incident modifications executed within a trigger script are only saved to the database after the modifications are completed.

## Incident De-Duplication

In the lifecycle of incident management, there are cases when incidents are duplicated. Cortex XSOAR provides the following de-duplication capabilities:

- **Manual De-Duplication:** You can manually de-duplicate incidents from the **Incidents** page or the **Related Incidents** page. To de-duplicate incidents manually, see [Manually De-Duplicate Incidents](#).
- **Automatic De-Duplication:** You can automate de-duplicate incidents by using [Pre-Process Rules](#) and [Scripts](#).
- **Automations:** You can create an automation that creates child incidents from duplicates.
- [Playbooks](#): Identify, review or close duplicate incidents using playbooks.

### Pre-Process Rules

Pre-Process rules enable you to perform certain actions on incidents as soon as they are ingested into Cortex XSOAR directly from the user interface. Through these rules, you can select incoming events on which to perform actions, for example, link the incoming incident to an existing incident, or under pre-configured conditions, drop the incoming incident altogether.

You can de-duplicate incidents by selecting the **Link and Close** action in the **Pre-Process Rules** tab. To create a pre-process rule, see [Create Pre-Process Rules for Incidents](#).

The **Link and Close** action creates an entry in the Linked Incidents table of the existing incident to which you link, and closes the incoming incident. If an existing incident matching the defining criteria is not found an incident is created for the incoming event.

### Playbooks

There are several out-of-the-box playbooks you can run to identify and close duplicate incidents. Alternatively, you can use these playbooks as the basis for customized de-duplication playbooks. For example, instead of automatically closing the duplicate incidents, include a manual review of the duplicate incidents.

| Playbook            | Description                                                                                                                                                                 |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dedup - Generic     | Identifies duplicate incidents using one of the supported methods, such as the machine learning model (used mainly for phishing).                                           |
| DeDup incidents     | Checks for duplicate incidents using the FindSimilarIncidents script, which is a rule-based script.<br><br>If duplicate incidents are found, they are closed as duplicates. |
| DeDup incidents -ML | You can set the threshold for the duplicate incidents. If duplicate incidents are found, they are closed as duplicates.                                                     |

## Automatic De-Duplication Using Scripts

There are various scripts you can use in automations and playbooks to identify and close duplicate incidents:

- FindSimilarIncidentsByText
- FindSimilarIncidents
- GetDuplicatesMI

### FindSimilarIncidentsByText

- Identifies similar incidents based on text similarity. For this script you specify incident keys, labels, or custom fields.
- The comparison is based on the [TF-IDF method](#).
- A score is calculated for each candidate (0-1), and incidents are considered duplicates when exceeding the threshold. The default threshold is 98%.

```
!FindSimilarIncidentsByText textFields=name,details
maximumNumberOfIncidents=1000 threshold=0.95 timeFrameHours=24
ignoreClosedIncidents=no
```

This command example checks for duplicate incidents using the following methodology:

1. Query for duplicate candidates:
  - Incidents created in the previous 24 hours [timeFrameHours=24].
  - Includes closed incidents [ignoreClosedIncidents].
  - Maximum number of incidents to check is 1,000 [maximumNumberOfIncidents=1000].
2. For each candidate, concatenate name and details incident fields [textFields=name,details] into a text document.
3. Compare the current incident text with all candidates using the TF-IDF method
4. Check if there is at least one similar candidate:
  - Candidates with a TF-IDF score of 95% [threshold=0.95]. If there is at least one candidate, announce duplicate.

---

## FindSimilarIncidents

- Rule-based script that identifies similar incidents based on common incident keys, labels, custom fields, or context keys.
- We recommend using incident keys, for example, "type" for same incident type.
- Due to performance considerations, we recommend not using context keys, for example, if the value also appears in the label key. Each duplicate candidate creates an additional server query.

```
!FindSimilarIncidents similarIncidentKeys="type,severity"
similarLabelsKeys="Email/from,Email/subject:*,Email/text:5"
ignoreClosedIncidents="yes" numberOfIncidents="1000" hoursBack="48"
timeField="created" maxResults="10"
```

This command example checks for duplicate incidents using the following methodology:

1. Query for duplicate candidates:
  - Incidents created in the 48 hours [hoursBack="48", timeField=created] before the original incidents
  - Excludes closed incidents [ignoreClosedIncidents=yes]
  - Maximum number of incidents to check is 1,000 [numberOfIncidents=1000]
  - Filters by the same incident type and severity [similarIncidentKeys=type,severity]
2. Check for candidate with the same Email/from label, or similar Email/subject label:
  - Contains, or contained, the original incident Email/subject label, and similar Email/text label
  - Equal or a maximum difference of 5 words from the original Email/text label [similarLabelsKeys="Email/from,Email/subject:\*,Email/text:5"]
3. If duplicate incidents are found, store the results in the context:
  - Maximum of 10 [maxResults="10"]

## GetDuplicatesMl

- Identifies duplicate incidents based on a machine learning (ML) algorithm, which uses ML techniques with predefined data. Alternatively, you can use data from the local environment.
- This script takes several features into consideration: labels comparison, email labels (relevant for phishing scenarios), incident time difference, and shared indicators (which you can customize with arguments).

```
!GetDuplicatesMl numberOfIncidents="1000" timeFrameDays="7"
ignoreClosedIncidents="yes" threshold="0.5" compareIndicators="Email,
IP, Domain, File SHA256, File MD5, URL" compareEmailLabels="Email/
headers/From, Email/headers/Subject, Email/text, Email/html, Email/
attachments" compareOtherLabels="yes" compareIncidentTimeDiff="yes"
UseLocalEnvDuplicatesInLastDays="0" ipComparisonSubnetMask="32"
maxCandidates="10"
```

This command example checks for duplicate incidents using the following methodology.

1. Query for duplicate candidates:
  - Incidents created in the 7 days [timeFrameDays="7"] before the original incidents
  - Excludes closed incidents [ignoreClosedIncidents=yes]

- 
- Maximum number of incidents to check is 1,000 [maxNumberOfIncidents=1000]
2. For each candidate calculate features based on similar email labels and other labels:
    - Email labels [compareEmailLabels, compareOtherLabels]
    - Indicators [compareIndicators, ipComparisonSubnetMask]
    - Time difference between the incidents [compareIncidentTimeDiff]
  3. Build machine learning model based on pre-defined data set:
    - Does not take into account local environment data set - linked and duplicate incidents in the system [UseLocalEnvDuplicatesInLastDays=0]
  4. Predict whether each candidate is a duplicate:
    - Prediction is based on a score (probability) between 0-1

## Manually De-Duplicate Incidents

You can manually de-duplicate incidents from the **Incidents** page or the **Related Incidents** tab.

- To de-duplicate incidents from the **Incidents** page:
  1. Select the incident check boxes that are duplicates.
  2. Click **Mark as Duplicate**.
  3. Type the ID of the incident that you want to retain.
  4. Click **Validate**.
  5. Click **Close others as duplicate**.

The duplicated incident is removed from the table.

- To de-duplicate incidents from the **Related Incidents** tab:
  1. From the **Incidents** page, select the relevant incident check box.
  2. In the **Related Incidents** tab, click the related incident you want to de-duplicate.
  3. Click **Duplicate**.
  4. Click **Close as duplicate**.

The duplicated incident is closed and marked as duplicate.

## Create Pre-Process Rules for Incidents

Pre-processing rules enable you to perform certain actions on incidents as they are ingested into Cortex XSOAR. Using the rules, you can select incoming events on which to perform actions, for example, link the incoming incident to an existing incident, or under configured conditions, drop the incoming incident altogether.



*Rules are applied in descending order, and only one rule is applied per incident.*

**STEP 1** | Select **Settings > Integrations > Pre-Process Rules > New Rule**.

**STEP 2** | In the **Rule Name** field, type a name for the rule.

It is recommended that you give meaningful names that help you identify what the rule does when viewing the list of rules.

**STEP 3** | If you want the rule to apply to a specific incident, in the **Conditions for Incoming incident** section, click **Add filter** and set the incident field and value.

For example, if you know there is a phishing campaign, you can create a rule for email subject with a potential phishing as the value.

You can add multiple conditions within a filter and add multiple filters. For more information about filters, see [Filters Reference](#).

**STEP 4** | In the **Action** section, from the drop down list determine which action to take in the event that the incoming incident matches the rule.

**STEP 5** | Depending on the Action field, complete section 3.

For example, in the **Action** field, if you select **Drop and update**, in section 3, complete the **Update** field.

Section 3 enables you to link to an incoming event and update the incident depending on the selected filter. For information about the **Action** section and the fields in section 3, see [Rule Actions for Pre-Process Rules](#).

**STEP 6** | (Optional) To test the rules to ensure they are effective and efficient, click **Test**.

Testing is useful for ensuring that you are receiving the desired results before putting a rule in production. It is recommended that you provide an existing incident as a sample incident against which the rule can run.

**STEP 7** | Click **Save**.

The screenshot displays a rule configuration interface with three main sections:

- 1 Conditions for Incoming Incident**: A section for defining conditions. It includes a dropdown menu for "Email Subject", a filter type of "Includes (String)", and a text input field containing "[contains phishing]". Below this is a "+ Add filter" button.
- 2 Action**: A section for defining the action to be performed. It features a dropdown menu with "Drop and update" selected.
- 3 Update**: A section for defining update parameters. It includes a "Link to" dropdown set to "oldest incident", a "Created within the last" field set to "30" and "Days", and a checked "Search closed incidents" checkbox. Below this is an "AND" section with a dropdown for "Email Subject", a filter type of "Is identical (Incoming incident)", and a text input field containing "of existing incident" and "to incoming incident". A "+ AND" button is located below this section.

At the bottom right of the interface are three buttons: "Cancel", "Test", and "Save".

In most cases, in a phishing campaign, the email subject is similar. In section 1, we create a condition for incoming incidents that contains phishing in the email subject. For example, *this is a phishing email*.

As this is a campaign, we want to drop the incoming incident and link (update) it to an existing incident.

In Section 3, we want to tell Cortex XSOAR which incident to link (update) the incoming incident. In this example, we link the email subject to the oldest incident, (link to the first incident in the campaign) and to those email subjects that are identical to the incoming incident.

### Rule Actions for Pre-Process Rules

The following table describes the rule action for pre-process rules.

| Option          | Description                                                                                                                                                                                                                                                                        | Section 3                                                                                                                                                                                                                           |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Drop            | Drops the incoming incident and no incident is created.                                                                                                                                                                                                                            | None                                                                                                                                                                                                                                |
| Close           | Closes the incoming incident.                                                                                                                                                                                                                                                      | None                                                                                                                                                                                                                                |
| Drop and update | Drops the incoming event, and updates the Dropped Duplicate Incidents table of the existing incident that you define. In addition, a War Room entry is created. If an existing incident matching the defined criteria is not found, an incident is created for the incoming event. | <b>Update</b> <ol style="list-style-type: none"> <li>1. Determine if you want to update the newest or oldest incident within a time range.</li> <li>2. Select the incident you want to update together with the value.</li> </ol>   |
| Link            | Creates an entry in the Linked Incidents table of the existing incident to which you link.                                                                                                                                                                                         | <b>Link to</b> <ol style="list-style-type: none"> <li>1. Determine if you want to link to the oldest or newest incident within a time range.</li> <li>2. Select the incident you want to update together with the value.</li> </ol> |
| Link and close  | Creates an entry in the Linked Incidents table of the existing incident to which you link, and closes the incoming incident. If an existing incident matching the defined criteria is not found, an incident is created for the incoming event.                                    | <b>Link to</b> <ol style="list-style-type: none"> <li>1. Determine if you want to link to the oldest or newest incident within a time range.</li> <li>2. Select the incident you want to update together with the value.</li> </ol> |
| Run a script    | Select a script to run on the incoming incident.<br><br> <i>When you create a script, you need to add the preProcessing tag for the script to appear in the list of available scripts.</i>      | <b>Choose a script</b><br><br>From the drop down list, select the script to run on the incoming incident. Only scripts that were tagged <b>preProcessing</b> appear in the drop-down list.                                          |

## Post Processing for Incidents

After you remedy an incident, you may want to perform additional actions on the incident, such as closing a ticket in a ticketing system or sending out an email. You can create a post-processing script to cover these scenarios.



If a post-processing script returns an error, the incident does not close.

You need to [Create a Post-Processing Script](#) and then [Add a Post-Processing Script to the Incident Type](#).

### Arguments Exposed in the Post-Processing Script

These arguments are exposed in the post-processing script:

- closed (closed time)
- status
- openDuration
- closeNotes
- Custom fields are set at closure either explicitly (through the CLI) or implicitly (through Cortex XSOAR).

## Create a Post-Processing Script

This procedure describes how to create a post-processing script after an incident has been remedied.

**STEP 1** | Select **Automation > New Automation**.

**STEP 2** | Type a name for the post-processing script and click **Save**.

**STEP 3** | In the **Tags** field, from the drop down list select **Post-processing**.

**STEP 4** | Add fields as required.

**STEP 5** | Click **Save**.

**STEP 6** | [Add a Post-Processing Script to the Incident Type](#).

The following script example requires the user to verify all *To Do* tasks before closing an incident. Before you start, you need to configure a Cortex XSOAR REST API instance.

```
inc_id = demisto.incidents()[0].get('id')
tasks = list(demisto.executeCommand("demisto-api-get", {"uri": "/todo/"}).format(inc_id))[0]['Contents']['response']

if tasks:
 for task in tasks:
 if not task.get("completedBy"):
 return_error("Please complete all ToDo tasks before closing the incident")
 break
```

## Add a Post-Processing Script to the Incident Type

Before you start you need to [Create a Post-Processing Script](#). After you add a post-processing script to the incident type, all incident types will use the post-processing script.

**STEP 1** | Go to **Settings > Advanced > Incident Types**.

**STEP 2** | Select the incident type you want to add the post processing script.

STEP 3 | Click **Edit**.

STEP 4 | In the **Post process using** field, from the drop down list, select the script.

Edit Incident Type

Name \* Phishing Color

Default playbook Phishing Playbook - Manual x

Run playbook automatically

Auto extract incident indicators Use system default ?

Post process using Choose script v

SLA

VerifyingToDo

VerifyingToDo

Weeks

Set Reminder at

Cancel Save

STEP 5 | Click **Save**.

## Incident Access Control Configuration

You can limit access to incidents by:

- Limiting access to incidents by [restricting an investigation](#).

When you mark an investigation as Restricted, only the incident owner and the team associated with the incident can view the investigation. Other users cannot view or access the incident. You can always remove the Restricted configuration.

- Limiting actions in the incident table according to roles.

You can edit the incident table action by going to **Settings > USERS AND ROLES > Roles** and selecting the table actions for the role. For more information about defining roles, see [Define a Role](#).

- [Limit Access to Investigations using RBAC](#).

### *Limit Access to Investigations using RBAC*

When you define access to an investigation according to Cortex XSOAR's roles, only users with that role can view and access the investigation.

STEP 1 | In the Incident page, select the incident you want to restrict access.

STEP 2 | In the CLI, type `/incident_set roles=name of the role`.

STEP 3 | To check that the role was assigned to the incident, click the **War Room** tab.

STEP 4 | (Optional) For automations:

- 
- Use the `setIncident` command in a playbook.
  - Specify the roles that you want to have access to the incident investigation.

## *Restrict an Investigation*

You can restrict an investigation to the incident owner and the team associated with the investigation.

**STEP 1** | Do one of the following:

- Open the incident and select **Actions > Restrict incident**.  
To remove the restriction select **Actions > Permit incident**.
- In the CLI, type `/investigation_restrict id= id_ number`

**STEP 2** | (Optional) For Automation do the following:

- Use the `restrictInvestigation` command in a playbook.
- Specify the incident ID of the incident for which you want to restrict access.
- Set the `Restrict` argument to `True` to restrict the incident.
- Set the `Restrict` argument to `False` to remove restricted from the incident.



# Playbooks

- > Playbooks Overview
- > Manage Playbook Settings
- > Playbook Inputs and Outputs
- > Playbook Tasks
- > Extend Context
- > Generic Polling
- > Filters and Transformers
- > Common Scripts to use in Automations



---

# Playbooks Overview

Playbooks are at the heart of the Cortex XSOAR system. They enable you to automate many of your security processes, including, but not limited to handling your investigations and managing your tickets. You can structure and automate security responses that were previously handled manually. For example, you can use playbook tasks to parse the information in the incident, whether it be an email or a PDF attachment. You can interact with users in your organization using communication tasks, or remediate an incident by interacting with a 3rd party integration.

Playbooks have different task types for each of the actions you want to take along the way. There are manual tasks where an analyst might have to confirm information or escalate an incident, and there are conditional tasks with a loop to check if certain information is present so you can proceed with your investigation. The playbook tasks can open tickets in a ticketing system, such as Jira, detonate a file using a sandbox.

As you are building out your playbook, keep in mind the following:

- What actions do you need to take?
- Which conditions might apply along the way? Are these conditions manual or automatic?
- Do you need to include looping?
- Are there any time-sensitive aspects to the playbook?
- When is the incident considered remediated?

## Task Types

The answers to the above questions will determine what kind of task you will need to create. Playbooks support the following task types:

- **Standard tasks** - these range from manual tasks like creating an incident or escalating an existing incident, to automated tasks such as parsing a file or enriching indicators. Automated tasks are based on scripts that exist in the system. These scripts can be something that was created by you, the user, or come pre-packaged as part of an integration. For example, the `!file` command enables you to enrich a file using any number of integrations that you have installed in your system. Alternatively, the `!ADGetUser` command is specific to the Active Directory integration.
- **Conditional tasks** - these tasks are used as decision trees in your flow chart. For example, were indicators found. If yes, you can have a task to enrich them, and if not you can proceed to determine that the incident is not malicious. Or, you can use conditional tasks to check if a certain integration is available and enabled in your system. If it is, you can use that integration to perform an action, and if not, you can continue to a different branch in the decision tree.

Conditional tasks can also be used to communicate with users through a single question survey, the answer to which determines how a playbook will proceed.

- **Data collection** - these tasks are used to interact with users through a survey. The survey resides on an external site that does not require authentication, thereby allowing survey recipients to respond without restriction.

All responses are collected and recorded in the incident's context data, whether you receive responses from a single user or multiple users. This enables you to use the survey questions and answers as input for subsequent playbook tasks.



*You can collect responses in custom fields, for example, a Grid field.*

- 
- Section headers - these tasks are used to manage the flow of your playbook and help you organize your tasks efficiently. You create a Section Header task to group a number of related tasks under the Section Header, as you would items in a warehouse or topics in a book.

For example, in a phishing playbook, you would have different sections for the investigative aspect of the playbook, such as indicator enrichment, and the tasks for communication with the user who reported the phishing.

## Inputs and Outputs

Depending on the task type that you select, and the script that you are running, your playbook task has inputs and outputs.

Inputs are data pieces that are present in the playbook or task. The inputs are often manipulated or enriched and they produce outputs. Outputs are objects whose entries will serve the tasks throughout the playbook, and they can be derived from the result of a task or command. To learn more about inputs and outputs, see [Playbook Inputs and Outputs](#).

## Field Mapping

You can map output from a playbook task directly to an incident field. This means that the value for an output key populates the specified field per incident. This is a good alternative to using a task with a set incident command.

---

# Manage Playbook Settings

You can manage general playbook settings such as the name, who can edit and run the playbook, as well as for which incident types the playbook runs in the playbook settings.

**STEP 1 |** From the Playbooks page, click the playbook that you want to manage.

**STEP 2 |** Click **Settings**.

**STEP 3 |** Add the following settings as required.

1. In the **Roles** field, from the dropdown list, select the roles for which the playbook is available.
2. In the **Playbook triggers** section, select the incident types for which this playbook runs.

This overwrites the settings configured in the respective incident types. The playbook currently defined for the incident type is listed under **Triggered playbook**.

3. In the **Advanced** section, determine whether the playbook runs in quiet mode.

When **Quiet Mode** is selected, playbook tasks do not display inputs and outputs and do not auto-extract indicators.

Playbook tasks are not indexed so you cannot search on the results of specific tasks. All of the information is still available in the context data, and errors and warnings are written to the War Room. **Quiet Mode** is recommended for scenarios that involve a lot of information that might adversely affect performance, for example, processing indicators from threat intel feeds.

**Playbook Settings** x

**Basic**

Name: Extract And Process Indicators From File

Description: Fill playbook description ...

Roles: Add roles

Enabled

**Playbook triggers**

**Advanced**

Quiet Mode ?

**STEP 4 |** Click **Save**.

# Playbook Inputs and Outputs

Playbooks and tasks have inputs, which are data pieces that are present in the playbook or task. The inputs are often manipulated or enriched and they produce outputs. The inputs might come from the incident itself, such as the role to whom to assign the incident, or an input can be provided by an integration. For example, when an Active Directory integration is used in a task to extract a user's credentials.

Playbook Inputs and Outputs

From context data  From indicators

Inputs Outputs

Name Value

SrcIP

Get  
incident.src  
Where  
No filters applied  
Transformers  
No transformers applied

Description Mandatory

The source IP address from which the incident originated.

Name Value

DstIP

Get  
incident.dest  
Where  
No filters applied  
Transformers  
No transformers applied

Description Mandatory

+ Add Input Cancel Save

In the image above, we see a playbook that is triggered based on context data, meaning an incident. The first two inputs are the `SrcIP`, which comes from the `incident.src` key, and `DstIP`, which is retrieved from `incident.dst`.

In addition, the playbook itself creates output object whose entries serve the tasks throughout the playbook.

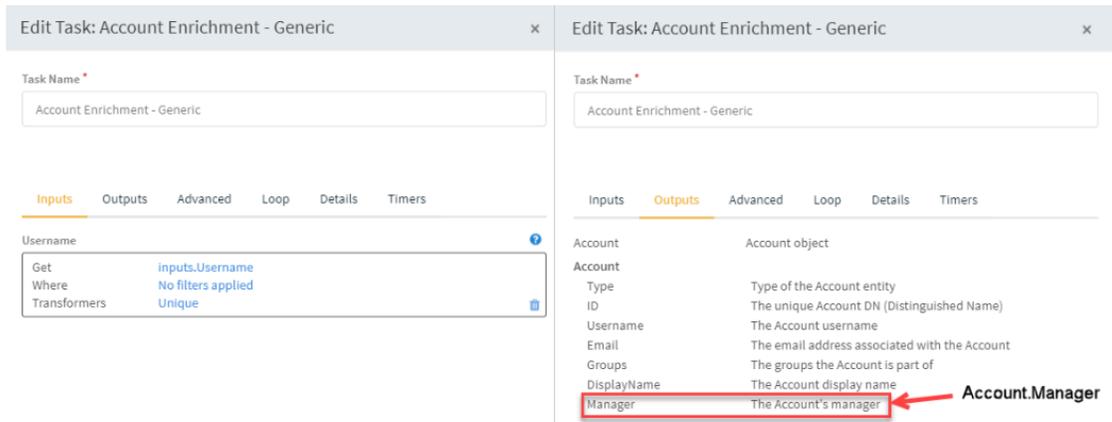
Playbook Inputs and Outputs
×

| Context path                                   | Description                                                 | Type      |    |
|------------------------------------------------|-------------------------------------------------------------|-----------|----|
| <input type="text" value="Endpoint"/>          | <input type="text" value="The Endpoint's object"/>          | Unknown ▾ | 🗑️ |
| <input type="text" value="Endpoint.Hostname"/> | <input type="text" value="The hostname to enrich"/>         | String ▾  | 🗑️ |
| <input type="text" value="Endpoint.OS"/>       | <input type="text" value="Endpoint OS"/>                    | String ▾  | 🗑️ |
| <input type="text" value="Endpoint.IP"/>       | <input type="text" value="List of endpoint IP addresses"/>  | Unknown ▾ | 🗑️ |
| <input type="text" value="Endpoint.MAC"/>      | <input type="text" value="List of endpoint MAC addresses"/> | Unknown ▾ | 🗑️ |
| <input type="text" value="Endpoint.Domain"/>   | <input type="text" value="Endpoint domain name"/>           | String ▾  | 🗑️ |

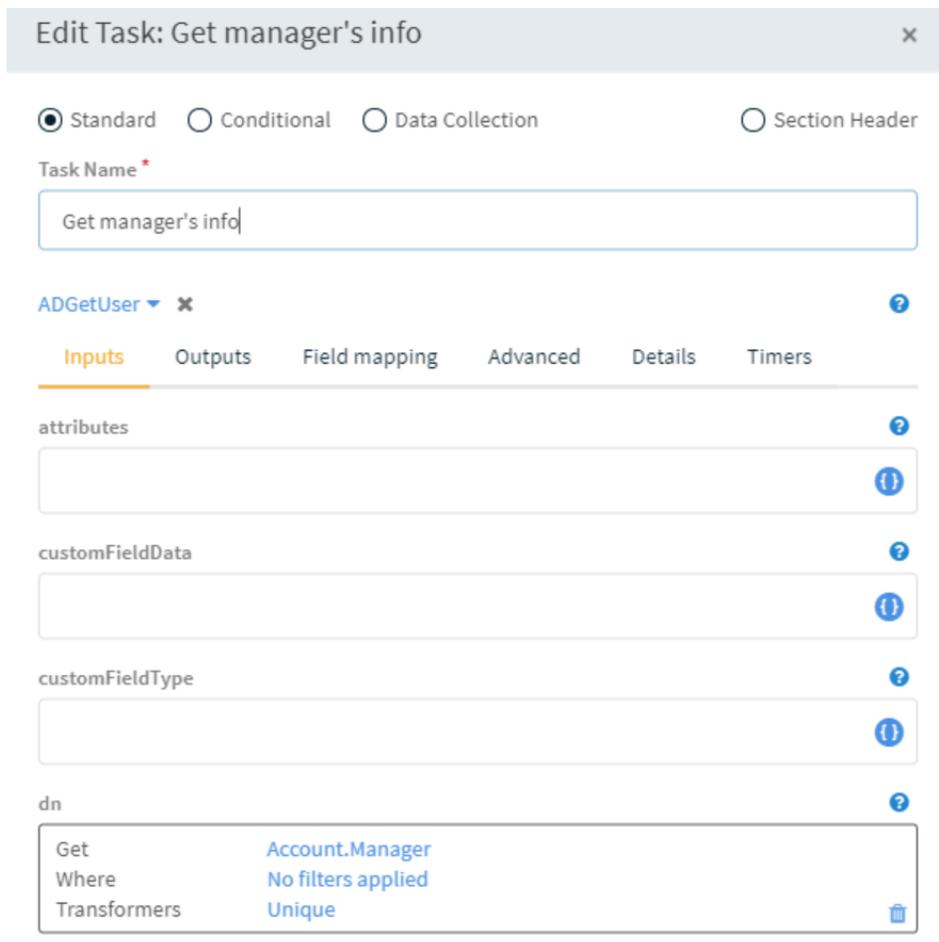
+ Add Output
Add manually
Cancel
Save

For example, we create a list of endpoint IP addresses which can later be enriched by an IP enrichment task, or a list of endpoint MAC addresses, which can be used to possibly get information about the hosts that were affected by the incident.

Outputs can also be data that was extracted or derived from the inputs. For example, in the following image we received the user's credentials from Active Directory, and used those credentials to retrieve the user's email address, manager, and any groups to which they belong.



An output can then serve as input for a subsequent task. For example, the user's manager who was returned as an output in the image above, can be used as an input to retrieve information from Active Directory.



Notice that the input for this task is Account.Manager, which is the output we highlighted in the playbooks inputs, above.

---

# Playbook Tasks

Cortex XSOAR supports different task types for the different aspects of the playbook. Each task type requires different information and provides different capabilities. You should choose your task type based on what you want to accomplish in the task. For example, for enrichment, you might want to run an enrichment sub-playbook or a command that returns additional information for an indicator. If you are at a fork in your decision tree, you should use a conditional task to help you determine which path to continue down.

- [Conditional Task](#)
- [Communication Tasks](#)

## Create a Conditional Task

Conditional tasks are used for determining different paths for your playbook. You can use conditional tasks for something simple like proceeding if a certain integration exists, or whether a user account has an email address.

Alternatively, you can use conditional tasks for more complex situations. For example, if an indicator was enriched and the reputation was set to bad, escalate the incident for managerial approval. However, if the indicator reputation is unknown or good, proceed down a different route.

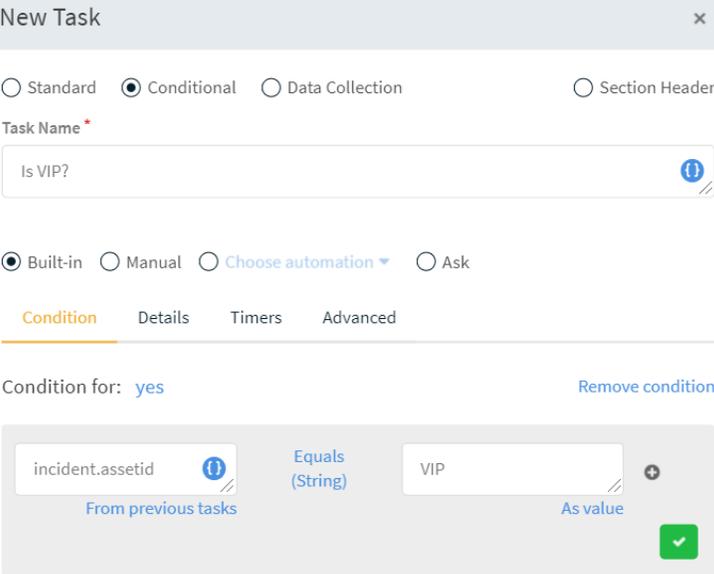
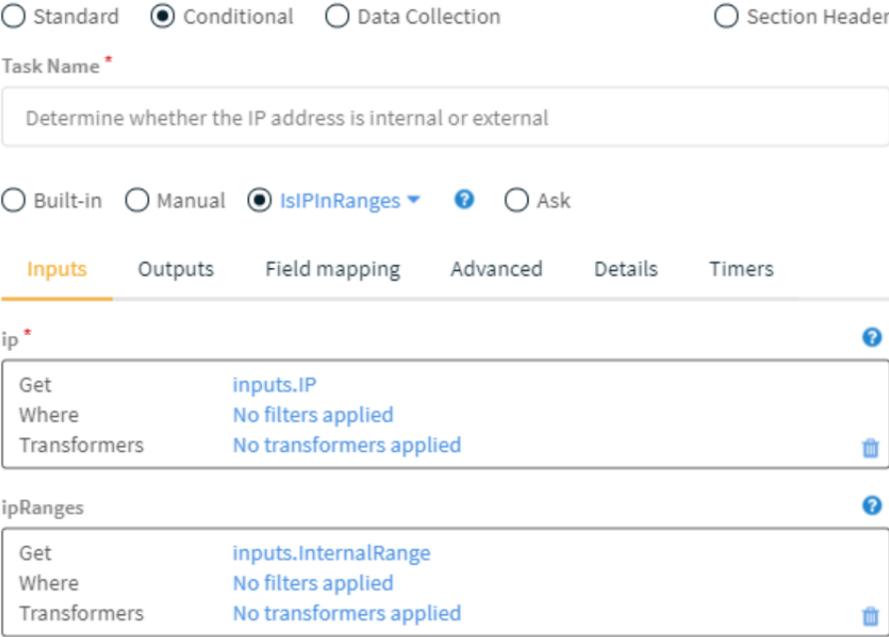
**STEP 1** | In a playbook, click + **Create Task**.

**STEP 2** | Select the **Conditional** option.

**STEP 3** | In the **Task Name** field, type a meaningful name for the task that corresponds to the data you are collecting.

**STEP 4** | Select the required option based on the conditional task.

| Option   | Description                                                                                                                                                                                                                                                                                          |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Built-in | Creates a logical statement using an entity from within the playbook. For example, in an access investigation playbook, you can determine that if the Asset ID of the person whose account was being accessed exists in a VIP list, set the incident severity to High. Otherwise, proceed as normal. |

| Option            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |     |           |       |                    |              |                         |     |                      |       |                    |              |                         |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----------|-------|--------------------|--------------|-------------------------|-----|----------------------|-------|--------------------|--------------|-------------------------|
|                   |  <p><b>New Task</b></p> <p> <input type="radio"/> Standard       <input checked="" type="radio"/> Conditional       <input type="radio"/> Data Collection       <input type="radio"/> Section Header     </p> <p>Task Name *</p> <p>Is VIP?</p> <p> <input checked="" type="radio"/> Built-in       <input type="radio"/> Manual       <input type="radio"/> Choose automation       <input type="radio"/> Ask     </p> <p>Condition Details Timers Advanced</p> <p>Condition for: yes <span>Remove condition</span></p> <p>incident.assetid <span>From previous tasks</span> Equals (String) VIP <span>As value</span></p>                                                                                                                                                                                                                                                                                                                                            |     |           |       |                    |              |                         |     |                      |       |                    |              |                         |
| Manual            | <p>Creates a conditional task which must be manually resolved. For example, in an access incident investigation, you might ask the user if they attempted to access their account. A manual task checks if the user responded.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |     |           |       |                    |              |                         |     |                      |       |                    |              |                         |
| Choose automation |  <p><b>New Task</b></p> <p> <input type="radio"/> Standard       <input checked="" type="radio"/> Conditional       <input type="radio"/> Data Collection       <input type="radio"/> Section Header     </p> <p>Task Name *</p> <p>Determine whether the IP address is internal or external</p> <p> <input type="radio"/> Built-in       <input type="radio"/> Manual       <input checked="" type="radio"/> IsIPInRanges       <input type="radio"/> Ask     </p> <p>Inputs Outputs Field mapping Advanced Details Timers</p> <p>ip *</p> <table border="1"> <tr> <td>Get</td> <td>inputs.IP</td> </tr> <tr> <td>Where</td> <td>No filters applied</td> </tr> <tr> <td>Transformers</td> <td>No transformers applied</td> </tr> </table> <p>ipRanges</p> <table border="1"> <tr> <td>Get</td> <td>inputs.InternalRange</td> </tr> <tr> <td>Where</td> <td>No filters applied</td> </tr> <tr> <td>Transformers</td> <td>No transformers applied</td> </tr> </table> | Get | inputs.IP | Where | No filters applied | Transformers | No transformers applied | Get | inputs.InternalRange | Where | No filters applied | Transformers | No transformers applied |
| Get               | inputs.IP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |     |           |       |                    |              |                         |     |                      |       |                    |              |                         |
| Where             | No filters applied                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |     |           |       |                    |              |                         |     |                      |       |                    |              |                         |
| Transformers      | No transformers applied                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |     |           |       |                    |              |                         |     |                      |       |                    |              |                         |
| Get               | inputs.InternalRange                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |           |       |                    |              |                         |     |                      |       |                    |              |                         |
| Where             | No filters applied                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |     |           |       |                    |              |                         |     |                      |       |                    |              |                         |
| Transformers      | No transformers applied                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |     |           |       |                    |              |                         |     |                      |       |                    |              |                         |

**STEP 5 |** Complete the task configuration in the remaining tabs. Some configurations are required, and some are optional.

STEP 6 | Click **Save**.

## Communication Tasks

Communication tasks enable you to send surveys to users, both internal and external, to collect data for an incident. The collected data can be used for incident analysis, and also as input for subsequent playbook tasks. For example, you might want to send a scheduled survey requesting analysts to send specific incident updates, or send a single (stand-alone) question survey to determine how an issue was handled.

To allow users outside the Cortex XSOAR server network to access the communication task link, you need to [configure access to the communication task through an engine](#).

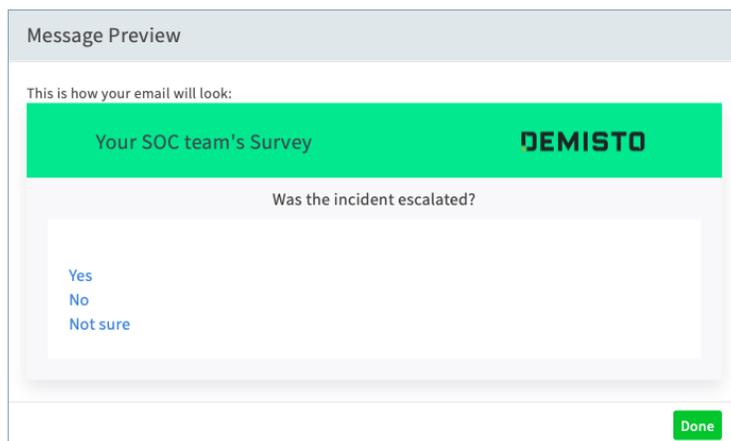
### Ask task

The conditional Ask task is a single question survey, the answer to which determines how a playbook will proceed. If you send the survey to multiple users, the first answer received is used, and subsequent responses are disregarded.

Users interact with the survey directly from the message, meaning the question appears in the message and they click an answer from the message.

The survey question and the first response is recorded in the incident's context data. This enables you to use this response as the input for subsequent playbook tasks.

Since this is a conditional task, it's important to remember to create a condition for each of the answers. For example, if the survey answers include, Yes, No, and Maybe, there should be a corresponding condition (path) in the playbook for each of these answers.



### Data Collection task

The Data Collection task is a multi-question survey (form) that survey recipients access from a link in the message. The survey resides on an external site that does not require authentication, thereby allowing survey recipients to respond without restriction.

All responses are collected and recorded in the incident's context data, whether you receive responses from a single user or multiple users. This enables you to use the survey questions and answers as input for subsequent playbook tasks.



*You can collect responses in custom fields, for example, a Grid field.*

---

## Create an Ask Task

The conditional Ask task is a single question survey, the answer to which determines how a playbook will proceed. If you send the survey to multiple users, the first answer received is used, and subsequent responses are disregarded.

Users interact with the survey directly from the message, meaning the question appears in the message and they click an answer from the message.

The survey question and the first response is recorded in the incident's context data. This enables you to use this response as the input for subsequent playbook tasks.

Since this is a conditional task, it's important to remember to create a condition for each of the answers. For example, if the survey answers include, Yes, No, and Maybe, there should be a corresponding condition (path) in the playbook for each of these answers.

In the task configuration there are several tabs that you can enter values for. Some configurations are required, and some are optional. For detailed information for each configuration tab.

**STEP 1** | In a playbook, click + **Create Task**.

**STEP 2** | Select the **Conditional** option.

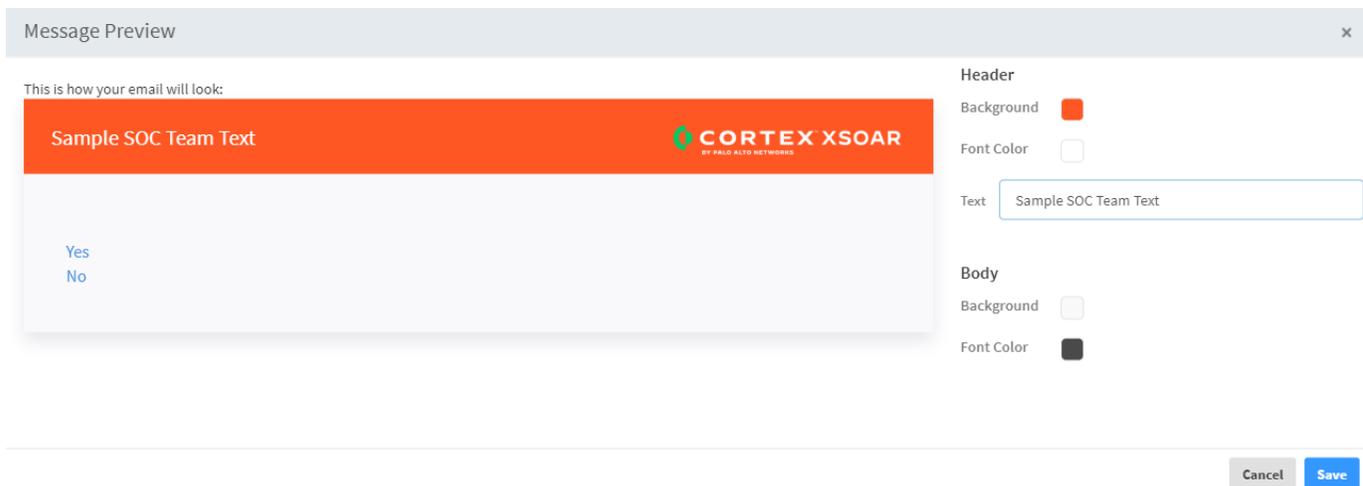
**STEP 3** | Enter a meaningful name for the task, which corresponds to the data that you are collecting.

**STEP 4** | Select the **Ask** option.

The screenshot shows the 'New Task' configuration window. At the top, there are four radio buttons: 'Standard', 'Conditional' (selected), 'Data Collection', and 'Section Header'. Below this is a 'Task Name' field containing 'Was the incident escalated?'. Underneath, there are four radio buttons for automation: 'Built-in', 'Manual', 'Choose automation' (with a dropdown arrow), and 'Ask' (selected). Below the automation options are five tabs: 'Message' (selected), 'Timing', 'Advanced', 'Details', and 'Timers'. The 'Message' tab contains the following fields: 'Ask by Email' (with a 'Preview' link), 'To' (a dropdown menu with 'Select from predefined values or add your own'), 'Subject' (an empty text field), and 'Message body' (an empty text area). Below these is the 'Reply Options' section with a plus icon and a text input field containing 'Yes'. At the bottom right, there are 'Cancel' and 'Save' buttons.

**STEP 5** | **Optional** To customize the look and feel of your email message, click **Preview**.

You can determine the color scheme and how text in the message header and body appear.

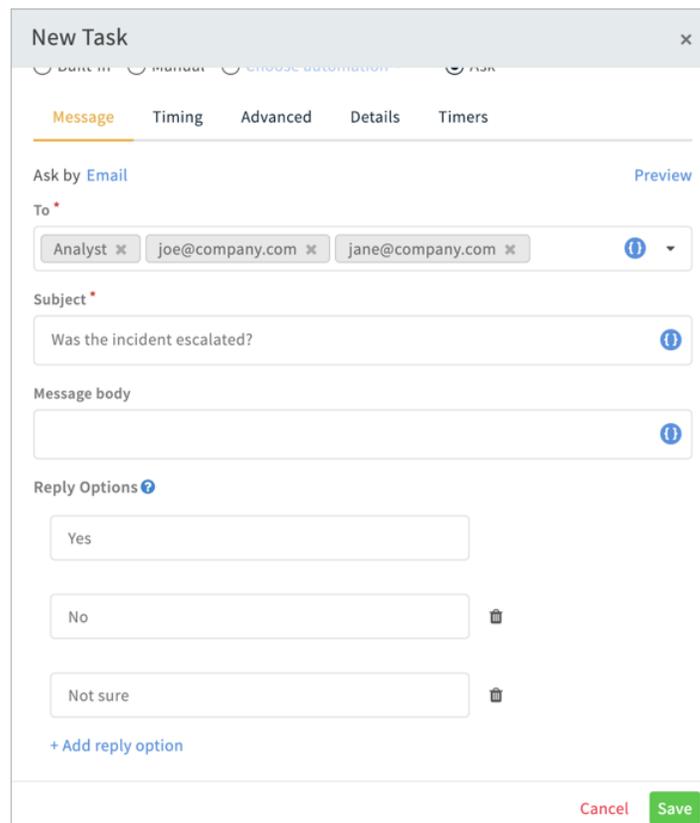


## Ask Task Examples

These examples show you how to implement some common uses of the Ask task.

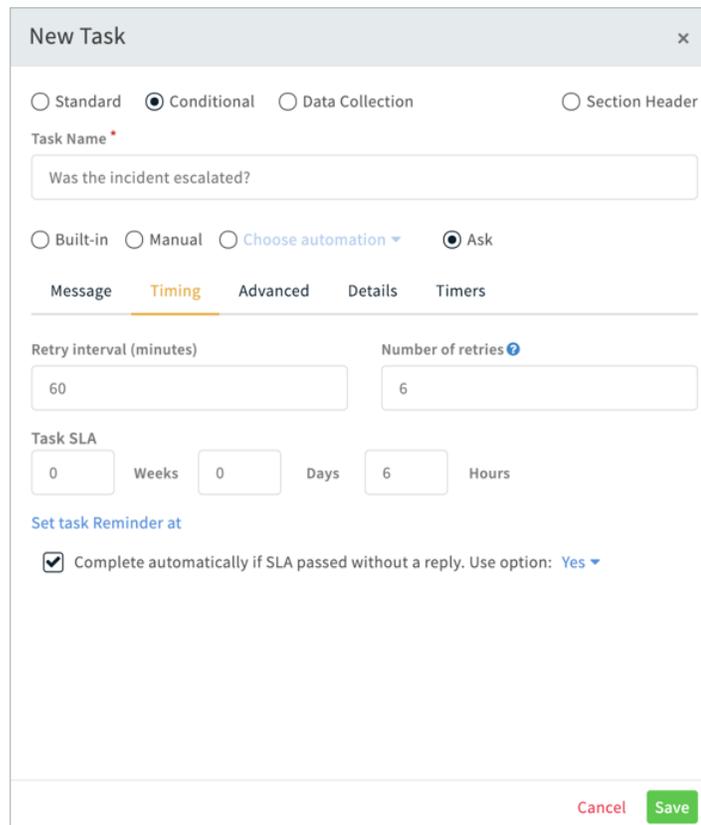
### *Send emails to users*

In this example, the message and survey are sent by email to all users with the Analyst role, and several external users. We didn't include a message body because the message subject is the survey question we want recipients to answer. There are three reply options, **Yes**, **No**, and **Not sure**. In the playbook, we will only add conditions for the Yes and No replies.



## Send a survey

In this example, the message and survey will be sent to recipients every hour for six hours, until a reply is received. The SLA is six hours. If the SLA is breached, the playbook will proceed according to the Yes condition.



### Customize an Ask Task

You can customize the message HTML and CSS for a data collection task.

**STEP 1** | Go to **Settings > About > Troubleshooting > Server Configuration**.

**STEP 2** | Add the key `messages.html.formats.externalAskSubmit`, and add the following HTML with your customizations as the value.

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml" xmlns:o="urn:schemas-microsoft-com:office:office">
 <head>
 <title> {{.title}} </title>
 <!--[if !mso]><!-- --> <meta http-equiv="X-UA-Compatible"
 content="IE=edge" <!--<![endif]--> <meta http-
 equiv="Content-Type" content="text/html; charset=UTF-8"> <meta
 name="viewport" content="width=device-width, initial-scale=1">
 <style type="text/css">
 #outlook a { padding: 0;
 text-decoration: none !important; color: #4da9ff;
 }
 a { text-decoration: none !
 important; color: #4da9ff; }
 .ReadMsgBody { width: 100%; }
 .ExternalClass { width: 100%; }
 .ExternalClass * { line-height: 100%; }
 body { margin: 0;
 padding: 0; -webkit-text-size-adjust: 100%; -ms-
 text-size-adjust: 100%; }
 table,
```

```

td { border-collapse: collapse; mso-table-
lspace: 0pt; mso-table-rspace: 0pt; }
img { border: 0; height: auto; line-
height: 100%; outline: none; text-decoration: none;
-ms-interpolation-mode: bicubic; }
p { display: block; margin: 13px 0;
}
</style> <!--[if !mso]><!--> <style
type="text/css"> @media only screen and (max-width:480px)
{ @-ms-viewport { width: 320px; } }
@viewport { width: 320px; } }
</style> <!--<![endif]--> <!--[if mso]>
<xml> <o:OfficeDocumentSettings>
<o:AllowPNG/> <o:PixelsPerInch>96</o:PixelsPerInch>
</o:OfficeDocumentSettings> </xml> <!
[endif]--> <!--[if lte mso 11]> <style type="text/
css"> .outlook-group-fix { width:100% !important; }
</style> <![endif]--> <style
type="text/css"> @media only screen and (min-width:480px)
{ .mj-column-per-100 { width: 100% !important;
} } </style>
<style type="text/css"> </style> <style type="text/
css"> div { margin: 0 auto;
} td { padding: 0;
} p { padding: 0; margin: 0;
} </style> </head> <body>
<div style="background-color: #f9f9fb;width:100%; height: 100%; font-
family: Helvetica, 'Trebuchet MS', Verdana, Geneva, Century Gothic, Arial,
sans-serif;"> <table style="background-color: #00eb9a; width: 100%;
height: 80px; vertical-align: middle;"> <tbody> <tr>
<td style="width: 70%;"> <h2 style="margin: 0 10px;">Message
From {{.title}}</h2> </td> <td style="width: 30%; text-
align: center;"></td> </tr> </tbody>
</table> <h3 style="text-align: center;">{{.subject}}</h3>
<div style="margin: 20px; background-color: white; padding: 20px;">
<p style="word-break: break-all;">{{.body}}</p>

{{.links}} </div> </div> </body> </html>

```

## Create a Data Collection Task

The Data Collection task is a multi-question survey (form) that survey recipients access from a link in the message. The survey resides on an external site that does not require authentication, thereby allowing survey recipients to respond without restriction.

All responses are collected and recorded in the incident's context data, whether you receive responses from a single user or multiple users. This enables you to use the survey questions and answers as input for subsequent playbook tasks.

You can include two types of questions in the survey; stand-alone questions and questions based on a Cortex XSOAR field.

- Stand alone questions are presented to users directly in the message, and from which users answer directly in the message (not an external survey).
- Field-based questions are based on a specific Cortex XSOAR field (either system or custom), for example, a Grid field. Questions The response (data) received for these fields automatically populates the field for this incident in Cortex XSOAR.

You can collect responses in custom fields, for example, a Grid field.



If responses are received from multiple users, data for multi-select fields and grid fields are aggregated. For all other field types, the most recent received response will override previous responses as it displays in the field. All responses are always viewable in the context data.

**STEP 1** | In a playbook, click + **Create Task**.

**STEP 2** | Select the **Data Collection** option.

**STEP 3** | Enter a meaningful name for the task that corresponds to the data you are collecting.

**STEP 4** | Determine how the message will appear to users and how the message or survey will be sent.

The survey does not appear in the message. A link to the survey is automatically placed at the bottom of the message.

**STEP 5** | Enter the questions that the survey will contain.

You create questions in the survey, you can drag-and-drop questions to rearrange the order in which they display in the survey.

**STEP 6** | **Optional** To customize the look and feel of your email message, click **Preview**.

You can determine the color scheme and how text in the message header and body appear, as well as the appearance and text of the button the user clicks to submit the survey.

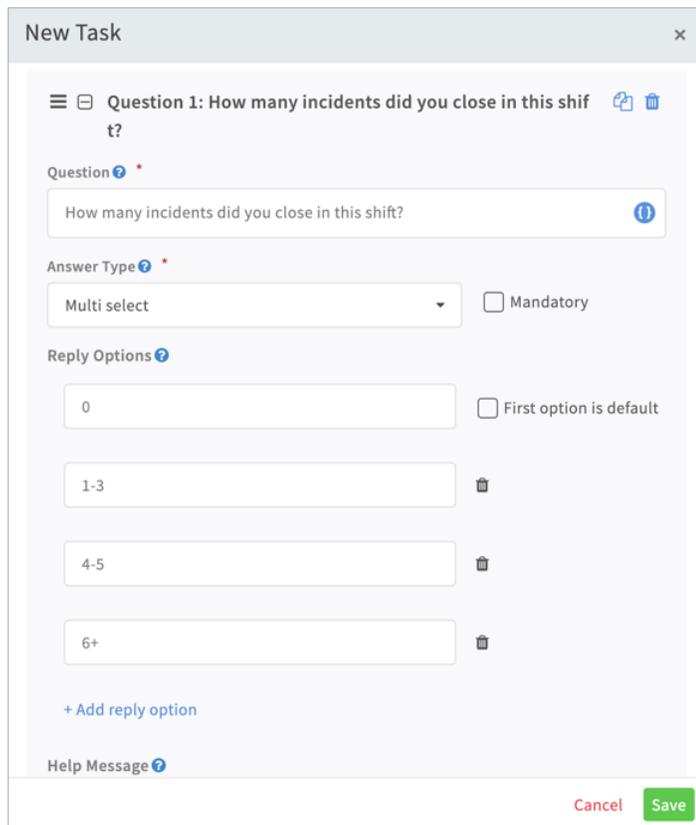
The screenshot displays the 'Message Preview' window, which is divided into two main sections: 'This is how your email will look:' and 'This is how the web form will look:'. The email preview shows a dark header with 'Your SOC team' and the 'CORTEX XSOAR' logo, followed by a body with a placeholder 'your link will be placed here.'. The web form preview shows a similar header, a body with 'your questions will appear here', a green 'Submit Answers' button, and a footer with the XSOAR logo and 'Powered by Cortex.XSOAR'. On the right side, there is a customization panel with sections for 'Header', 'Body', and 'Button'. Each section has options for 'Background' (color swatches), 'Font Color' (color swatches), and 'Text' (input fields). The 'Header' text field contains 'Your SOC team'. The 'Button' text field contains 'Submit Answers'. At the bottom right of the preview window, there are 'Cancel' and 'Save' buttons.

## Data Collection Task Examples

These examples show you how to implement some common uses of the Data Collection task.

### Stand-alone with multi-select answer

In this example, we created a stand-alone question, with a multi-select answer. Note that this question is not mandatory, and we did not select the **First option is default** checkbox. Had we selected this checkbox, the Reply Option "0" would be the default value in the answer field.



The screenshot shows a 'New Task' configuration window. At the top, it displays 'Question 1: How many incidents did you close in this shift?'. Below this, the question text 'How many incidents did you close in this shift?' is shown in a text field. The 'Answer Type' is set to 'Multi select', and the 'Mandatory' checkbox is unchecked. Under 'Reply Options', there are four input fields: '0', '1-3', '4-5', and '6+'. The 'First option is default' checkbox is also unchecked. A '+ Add reply option' link is visible below the options. At the bottom right, there are 'Cancel' and 'Save' buttons.

### Field-based using a custom field

In this example, we created a question based on a custom Grid field that we marked as mandatory. For the question field, we included a descriptive sentence explaining how to fill in the grid.

**Edit Task: Analyst Status Report for Shift**

6+

+ Add reply option

Help Message ?

☰ Question 2: Please detail the Closed Incidents for your shift. 🔗 🗑️

Question ? \* Detach from field

Please detail the Closed Incidents for your shift. ℹ️

Field associated with this question ? \*

Analyst Status Report: Closed Incidents  Mandatory

Help Message ?

+ Add Question + Add Question based on field ? Top of page

Cancel OK

### Customize a Data Collection Task

You can customize the message HTML and CSS for a data collection task.

**STEP 1** | Go to **Settings > About > Troubleshooting > Server Configuration**.

**STEP 2** | Add the key `messages.html.formats.externalFormSubmit`, and add the following HTML with your customizations as the value.

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:VML" xmlns:o="urn:schemas-microsoft-com:office:office">
 <head>
 <title> {{.title}} </title>
 <!--[if !mso]><!--
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <!--<![endif]>-->
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <style type="text/css">
 #outlook a { padding: 0; text-decoration: none !important; color: #4da9ff; }
 a { text-decoration: none !important; color: #4da9ff; }
 .ReadMsgBody { width: 100%; }
 .ExternalClass { width: 100%; }
 .ExternalClass * { line-height: 100%; }
 body { margin: 0; padding: 0; -webkit-text-size-adjust: 100%; -ms-text-size-adjust: 100%; }
 table, td { border-collapse: collapse; mso-table-lspace: 0pt; }
 img { border: 0; height: auto; line-height: 100%; outline: none; text-decoration: none; }
 p { interpolation-mode: bicubic; display: block; margin: 13px 0; }
 </style>
```



Name	Description
	tasks until a user with an external email address completes the task.
Set task reminder	Define a reminder for the task, in weeks, days, or hours.

## Field Mapping

Map output from a playbook task directly to an incident field.



*The output value is dynamic and is derived from the context at the time that the task is processed. As a result, parallel tasks that are based on the same output, might return inconsistent results.*

1. In the **Field mapping** tab, click **Add custom output mapping**.
2. Under **Outputs**, select the output parameter whose output you want to map. Click the curly brackets to see a list of the output parameters available from the automation.
3. Under **Field to fill**, select the field that you want to populate with the output.
4. Click **Ok**.

## Advanced Fields

Name	Description
Using	Determine which integration instance processes the script you select for this task.
Extend context	Determine which information from the raw JSON you want to add to the Context Data. This must be entered as contextKey=RawJsonOutputPath.
Ignore outputs	When selected, this takes the results from the Extend context field and overwrites existing output.
Execution timeout	Define how long a command waits, in seconds, before it times out.
Only the assignee can complete the task	Stop the playbook from proceeding until the task assignee completes the task. By default, in addition to the task assignee, the default administrator can also complete the blocked task. You can also block tasks until a user with an external email address completes the task.
Number of retries	Determine how many times the script attempts to run before generating an error.
Retry interval	Determine the wait time (in seconds) between each execution of the script.

Name	Description
Auto extract indicators	<p>Determines whether or not indicators from this task should be automatically extracted, and if so, using which method. Valid values are: Use system default - use the option defined in the system configuration. For more information, see Auto Extract. None - Indicators are not automatically extracted. Use this option when you do not want to automatically extract and enrich the indicators. Inline - Indicators are extracted and enriched within the task. Use this option when you need to have the most robust information available per indicator.</p> <p> <i>This configuration slows down your system performance.</i></p> <p><i>Out of band - Indicators are enriched in parallel (or asynchronously) to other actions. The enriched data is available within the incident, however, it is not available for immediate use in task inputs or outputs since the information is not available in real time.</i></p>
Mark results as note	<p>Select to make the task results available as a note. Notes are viewable in the War Room.</p>
Mark results as evidence	<p>Select to make the task results available as evidence. Evidence is viewable in the War Room.</p>
Run without a worker	<p>Select to execute this task without requiring a worker. When cleared, this task will only execute when there is a worker available.</p>
Skip this branch if this automation/playbook is unavailable	<p>Select to enable the playbook to continue executing if an instance of the automation, playbook, or sub-playbook is not available.</p>
Quiet Mode	<p>Determine if this task operates in Quiet Mode. When in Quiet Mode, tasks do not display inputs and outputs, nor do they auto-extract indicators. Errors and Warnings are still documented. You can determine to turn Quiet Mode on or off for a given task or control Quiet Mode by what is defined at the playbook level.</p>

---

## Details Fields

Name	Description
Tag the result with	Add a tag to the task result. You can use the tag to filter entries in the War Room.
Task description	Provide a description of what this task achieves.

## Timers Fields

Name	Descriptions
Timer action	Determine which action to take when the timer is triggered. Valid values are: Start, Stop, and Pause.
Select timer field	Select the field on which the timer is applied.

## Message Body Fields

Field	Description
Ask by	The method for sending the message and survey. EmailSlackMattermostIf you do not specify this parameter, the Allow from workplan method will be enforced, meaning users can complete the survey from the workplan.
To	<p>The message and survey recipients. There are several ways to define the recipients.</p> <p><b>User role:</b> Click inside the field to select a user role. All users assigned to the role will receive the message and survey.</p> <p><b>Email address:</b> Manually type email addresses for Cortex XSOAR users and/or external users.</p> <p><b>Context:</b> Click the context icon to define recipients from context data.</p>
Subject	The message subject that displays to message recipients. You can make the survey question the subject, but if you don't write the question here, you should write the question in the message body field.
Message body	The text that displays in the body of the message. Although this field is optional, if you don't write the survey question in the Subject field, you should include it in the message body. This is a long-text field.

Field	Description
Reply Options	The answers that display in the message, which users can select directly from the message.
Set task reminder	The schedule, in weeks, days, or hours, to resend the message and survey to recipients before.

## Timing Fields

The configuration options in the Timing tab define the frequency that the message and survey are resent to recipients before the first response is received, and the task SLA.

Field	Description
Retry interval	Determine the wait time between each execution of a command. For example, the frequency (in minutes) that a message and survey are resent to recipients before the response is received.
Number of retries	Determine how many times a command attempts to run before generating an error. For example, the maximum number of times a message is sent. If a reply is received, no additional retry messages will be sent.
Task SLA	Define the deadline for the task, in weeks, days, or hours.
SLA Breach	Select this checkbox to complete the task if the SLA is breached before a reply is received. You can select which condition is applied to continue the playbook.

## Questions Fields

### Stand-alone questions

Field	Description
Question	A question to ask recipients.
Answer Type	The field type for the answer field. Valid values are: Short text Long text Number Single select - requires you to define a reply option. Multi select - requires you to define a reply option. Date picker Attachments
Mandatory	If this checkbox is selected for a question, survey recipients will not be able to submit the survey until they answer this question.

Field	Description
Help Message	The message that displays when users hover over the question mark help button for the survey question.
Placeholder	The empty value text that displays in the question's answer field.

### Field-based questions

Field	Description
Question	The question that displays before the field for users to complete. This field doesn't necessarily need to be a question, it can also be a descriptive sentence explaining how users should complete the field.
Field associated with this question	The field associated with the question will automatically take all the parameters from the field definition, unless otherwise defined.
Mandatory	If this checkbox is selected for a question, survey recipients will not be able to submit the survey until they answer this question.
Help Message	The message that displays when users hover over the question mark help button for the survey question.

---

# Extend Context

There are situations where the data returned by an integration or command is not exactly what you need. For example, when you run the command `!ad-get-user name="sampleName"`, you receive information about the user's dn, email, and more. However, you might only want the name of the user's manager. And you might want to store that information in a specific field.

Cortex XSOAR enables you to do that using the Extend Context feature. Extend Context can be used as in the situation above, or when you want to run a command multiple times and save the output to a different key each time. Using our `!ad-get-user` command from above, run the command once to retrieve the user, and once to retrieve the user's manager. Also, there are situations where an integration is configured to display only some of the information that you retrieved. The information is shown in context-data, but there is more available. For example, when you run a command to retrieve offenses from a SIEM, only some of the information is displayed, per the configuration of the instance. You can use Extend Context to retrieve the additional information and place it in a new field.

By default, when you run a command, either from the command line or as part of a script or playbook, a subset of JSON fields are returned. To display the full JSON response, run the command using the `raw-response=true` flag.

## Extend Context in a Playbook Task

You can extend context either in a playbook task, or directly from the command line. Whichever method you use, Cortex XSOAR recommends that you first run your command with the `raw-response=true` flag. This will help you identify the information that you want to add to your extended data.

**STEP 1** | Navigate to **Advanced** tab of the relevant playbook task.

**STEP 2** | In the **Extend Context** field, enter the name of the field in which you want the information to appear and the value you want to return. For example, using the `ad-get-user` command, you could enter `user=attributes.displayName` to place the user's name in the `user` key.

To include more than one field, separate the fields with a double colon. For example:  
`user=attributes.displayName::manager=attributes.manager`

**STEP 3** | To output only the values for Extend context and ignore the standard output for the command, select the **Ignore Outputs** checkbox.

While this will improve performance, only the values that you request in the **Extend Context** field will be returned. In addition, you cannot use Field Mapping as there is no output to which to map the fields.

## Extend Context using the Command Line

You can extend context either in a playbook task, or directly from the command line. Whichever method you use, Cortex XSOAR recommends that you first run your command with the `raw-response=true` flag. This will help you identify the information that you want to add to your extended data.

**STEP 1** | Run your command with the extend-context flag `!<commandName> <argumentName> <value> extend-context=contextKey=JsonOutputPath`.

For example, to add the user and manager fields to context use the `ad-get-user` command, as follows:

```
!ad-get-user username=${user.manager.username} extend-
context=manager=attributes.manager::user=attributes.displayName
```

---

**STEP 2** | To output only the values that you set as Extend context, run the command with the ignore-ouput flag=true. `!ad-get-user username=${user.manager.username} extend-context=manager=attributes.manager::user=attributes.displayName ignore-output=true`

### Example

By default, offenses pulled from QRadar to Cortex XSOAR return 11 fields, including event count, offense type, description, and more. In the following example, we use extended context to show which additional information is available and how to map it to a field:

- Run the command `!qradar-offenses raw-response="true"`. You see that there are an additional 20 fields or so that are retrieved.
- Identify the fields that you want to add and run your command. For example, to retrieve the number of devices affected by a given offense, as well as the domain in which those devices reside, run the following command: `!qradar-offenses extend-context=device-count=device_count::domain-id=domain_id`

---

# Generic Polling

When working with certain 3rd party products (detonation, scan, search, etc.) you may have to wait for a process to finish on the remote host before continuing. In those cases, the playbook should stop and wait for the process to complete on the 3rd party product, and continue when it is done. You may not achieve this via integrations or automations due to hardware limitations. One method for solving this is using the **GenericPolling** playbook.

The **GenericPolling** playbook periodically polls the status of a process being executed on a remote host, and when the host returns that the process execution is done, the playbook finishes execution.

## How to use

Follow these instructions to use the **GenericPolling** playbook.

## Prerequisites

- **Start command:** The command that fetches the initial state of the process and save it to the context. This command usually starts the process that should be polled. For example:

Detonation: Submits a sample for analysis (detonated as part of the analysis). For example, **joe-analysis-submit-sample**.

Scan: Starts a scan for specified asset IP addresses and host names. For example, **nexpose-start-assets-scan**

Search: Searches in QRadar using AQL. For example, **qradar-searches**.

- **Polling command:** The command that polls the status of the process and saves it to the context. The command input **must be checked** as **Is array**, as this allows the playbook to poll at once more than a single process being executed. For example:

Detonation: Returns the status of the analysis execution. For example, **joe-analysis-info**.

Scan: Returns the specified scan. For example, **nexpose-get-scan**.

Search: Gets a specific search id and status. For example, **qradar-get-search**

## Inputs

Input	Description
<b>Ids</b>	A list of process IDs to poll (usually a previous task output).
<b>PollingCommandName</b>	Name of the polling command to run.
<b>PollingCommandArgName</b>	Argument name of the polling command. The argument should be the name of the process identifier (usually an ID).
<b>dt</b>	Cortex XSOAR Transform Language filter to be checked against the polling command result. Polling will stop when no results are returned from the DT filter.
<b>Interval</b>	Interval between each poll (default is 1 minute).
<b>Timeout</b>	The amount of time that'll pass until the playbook will stop waiting for the process to finish. After this time has passed the playbook will finish

Input	Description
	running, even if it didn't get a satisfactory result (the action is done executing).
<b>AdditionalPollingCommandArgNames</b>	If the polling command has more than a single argument you can add their names via this input, for example: arg1,arg2,....
<b>AdditionalPollingCommandArgsValues</b>	If the polling command has more than a single argument you can add their values via this input for example: value1,value2,....

## Generic Polling Example

### Detonate File - JoeSecurity

The image shows the configuration for a 'GenericPolling' task and its execution flow. On the left, the configuration panel includes:

- Ids:** A list containing 'Joe.Analysis.WebID'.
- PollingCommandName:** 'joe-analysis-info'.
- PollingCommandArgName:** 'webid'.
- Interval:** 'inputs.Interval'.
- Timeout:** 'inputs.Timeout'.
- dt:** 'Joe.Analysis(val.Status != 'finished').WebID'.
- AdditionalPollingCommandArgNames:** (empty).

On the right, the workflow diagram in the 'Security' context shows the following steps:

- Decision: 'Is JoeSecurity?' (#7). If YES, proceed to step 2.
- Decision: 'Is there a File to detonate?' (#6). If YES, proceed to step 3.
- Action: 'JoeSecurity Upload File' (#1).
- Action: 'GenericPolling' (#2).
- Action: 'JoeSecurity Get Info' (#8).

- **Start command:** The `joe-analysis-submit-sample` command starts a new analysis of a file in Joe Security.
- **Polling command:** The `joe-analysis-info` command returns the status of the analysis execution.
- **Argument name:** The `webid` argument name of the polling command.
- **Context path to store poll results:** `Joe.Analysis`  
**ID context path:** `webid` stores the ID of the process to be polled.  
**Status context path:** `status` stores the status of the process.
- **Possible values returned from polling command:** `starting`, `running`, `finished`.
- **DT** We want a list of IDs of the processes that are still running. Let's explain how it's built:  
`Path.To.Object(val.Status != 'finished').ID` Get the object that has a status other than

---

'running', then get its ID field. The polling is done only once the result is **finished**. The dt filter returns an empty result in that case, which triggers the playbook to stop running.

## Limitations of Generic Polling

- **Global context** is not supported.
- Does not run from the **Playground**.
- The polling command must support a list argument.

Argument *	Mandatory	Default	Description
<input type="text" value="webid"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Web IDs, supports comma-seperated arrays.
Initial value			<input checked="" type="checkbox"/> Is array
<input type="text"/>			

---

# Filters and Transformers

In Cortex XSOAR, data is extracted and collected from various sources, such as playbook tasks, command results, and fetched incidents, which is presented in JSON format. The data can be manipulated by using filters and transformers. You can add filters and transformers in a Playbook task or when mapping an instance.

## Filters

Filters enable you to extract relevant data, which you can use elsewhere in Cortex XSOAR. For example, if an incident has several files with varying file types and extensions, you can filter the files by file extension or file type, and use the filtered files in a detonation playbook. To create a filter, see [create a filter](#).



*In Cortex XSOAR version 5.5 and below, if you want to filter more than one object such as nested objects you need to either [create a top level key using the Set command](#) or use a [transformer as a filter](#).*

## Transformers

Transformers enable you to take one value and transform or render it to another value. For example, converting a date in non-Unix format to Unix format. Another example is applying the `count` transformer, which renders the number of elements.

To create a transformer, see [create a transformer](#). When you have more than one transformer, they apply in the order that they appear. You can reorder them using click-and-drag.

## Create Filters and Transformers in a Playbook

You can create filters and transformers when adding or editing a task in a playbook or when mapping an instance.



*In Cortex XSOAR version 5.5 and below you cannot filter nested objects unless you use the [Set command](#), or use a [filter as a transformer](#).*

**STEP 1** | Create or edit a playbook task.

**STEP 2** | In the field you want to add a filter or transformer, click  and then select **Filters and Transformers**.

**STEP 3** | In the **Get** field, type or select data you want to filter or transform.

**STEP 4** | (Optional) To filter the data, click **Add filter**.

1. Select the data you want to filter.
2. Select the [Filter Operators](#).
3. Add the value.

For example, to filter for PDF file extensions:

Get **File.Name**

Filter, **File.Type** equals **PDF**.

4. Click the tick box to save the filter.

**STEP 5** | (Optional) To apply transformers to the field, click **Add transformer**.

1. Click the transformer and select the relevant transformer.  
For example, you may want to change the date format for when incidents occurred.
2. Select the [Transformers Operators](#).
3. Click the tick box to save.

**STEP 6 |** (Optional) To test the filter or transformation click **Test** and select the investigation or add it manually.

Test ×

Get	incident.occurred
Where	No filters applied
Transformers	TimeStampToDate

Data is taken from investigation #222 Sample Incident - Unclassified ▾

▼ root: {} 5 items Edit Clear

- ▶ subplaybook-11: {} 1 item
- ▶ subplaybook-43: {} 1 item
- ▶ subplaybook-48: {} 1 item
- ▶ subplaybook-52: {} 1 item
- ▶ incident: {} 65 items

**Test**

Test result

```
1970-01-01T00:33:40.000Z
```

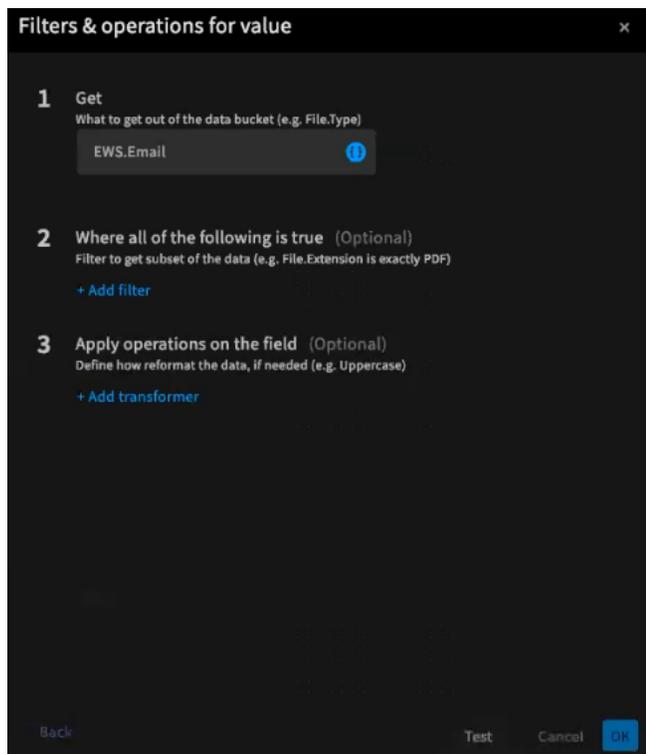
**Done testing**

## Filter Objects Using a Transformer

When filtering data, Cortex XSOAR only filters the first object in the data. If you want to filter nested data you can either [Filter Objects Using the Set Command](#) or use a transformer as a filter.

For example, to filter an email from **EWS.Email**, if you add a filter for an email, Cortex XSOAR filters EWS and not the Email. To filter the email, you need to add a transformer as a filter using the **WhereFieldEquals** operator.

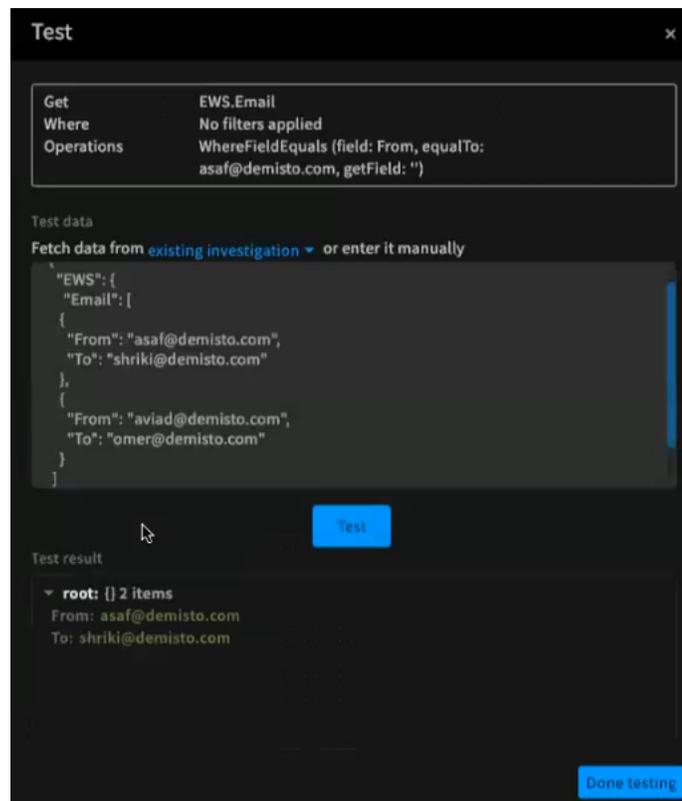
**STEP 1 |** In the **Get** field, type the data you want to use. For example, **EWS.Email**.



**STEP 2** | In the **Apply operations on the field**, add a transformer. In the this example, we want to filter all emails from **asaf@demisto.com**.



**STEP 3** | Click **Test** to view the filter.



## Filter Objects Using the Set Command

You can only filter top-level objects, not nested objects. To filter nested objects, you need to either [Filter Objects Using a Transformer](#) or add a top-level key by using the `set` command, which adds the nested context data to the top level. You can then create the filter.

For example, we have a number of URLs, and want to see only amazon.com. We need to add `URLData` to the top level key and then filter the context data for amazon.com.

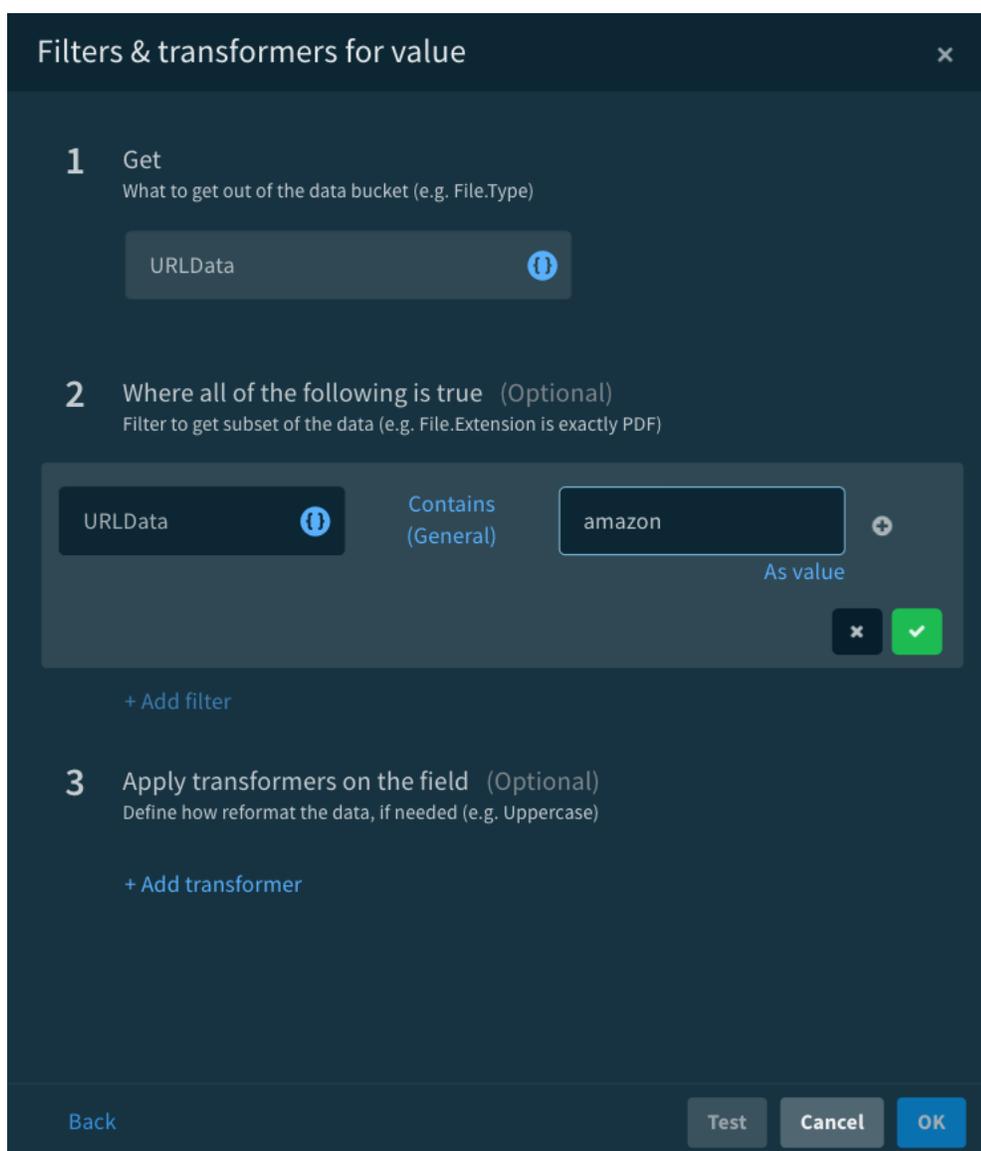
**STEP 1 |** Add the nested context data to a top level key by typing the following command in the Playground:

```
!set key=<data to add> value=<value of the data to add>
```

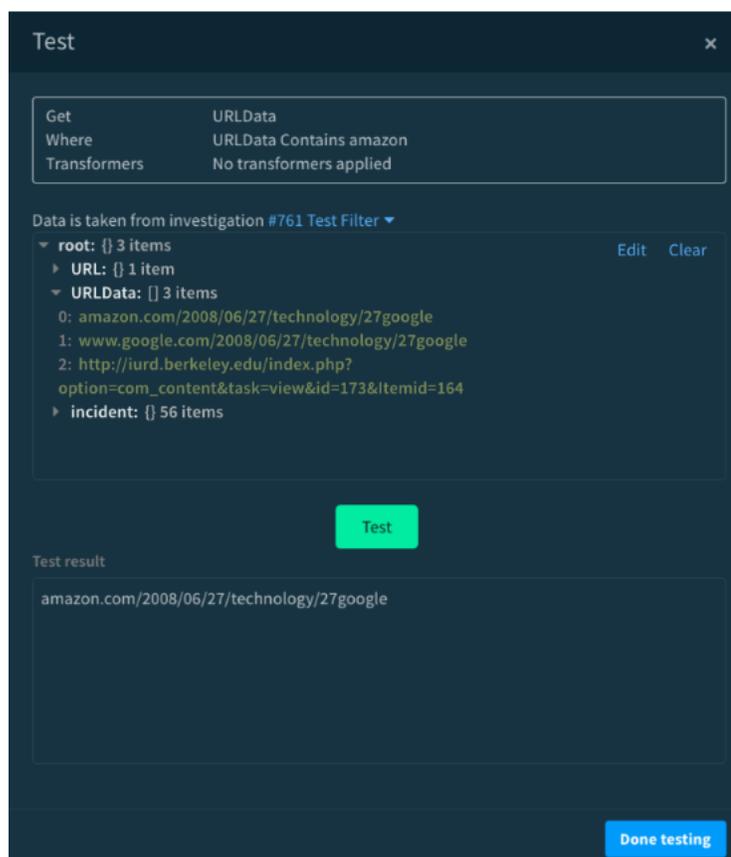
For example, to filter URL data, type `!set key=URLData value=$(URL.Data)`. You can see that `URLData` has been added to the top level:

```
▼ root: {} 3 items
 ▼ URL: {} 1 item
 ▼ Data: [] 3 items
 0: amazon.com/2008/06/27/technology/27google
 1: www.google.com/2008/06/27/technology/27google
 2: http://iurd.berkeley.edu/index.php?option=com_content&task=view&id=173&Itemid=164
 ▼ URLData: [] 3 items
 0: amazon.com/2008/06/27/technology/27google
 1: www.google.com/2008/06/27/technology/27google
 2: http://iurd.berkeley.edu/index.php?option=com_content&task=view&id=173&Itemid=164
```

**STEP 2** | Create the filter in the **Filters and transformers** window.



**STEP 3** | (Optional) Click **Test** to test the filter.



## Filter Operators

Filters enable you to extract relevant data for use elsewhere in Cortex XSOAR. For example, if an incident has several files with varying file types and extensions, you can filter the files by file extension or file type, and use the filtered files in a detonation playbook.

Note the following:

- Filters try to cast the transformed value and arguments to the appropriate type. The task fails if casting fails. For example, "a" Equals {"some": "object"} => Error
- If the filter's left-side value expects a single item, but receives a list, the filter passes if at least one item meets the requirements. For example, ["a", "b", "c"] Equals "b" => true.
- If the filter's left-side value expects a list, but receives a single item, it converts it to a list with a single item. For example, "a" Contains "a" => True.
- Some filters are implemented as automations, meaning custom transformers, automation with the filter tag. You can find examples in the automation description. For more information about creating custom filters, [Create Custom Filters and Transformers Operators](#).
- You can only filter at the root level. Filters cannot filter items nested in an object, such as a list, unless you [Filter Objects Using a Transformer](#) or [Filter Objects Using the Set Command](#).
- Filters in conditional tasks do not iterate the items of the root. Instead, they fetch the left-side value and the right-side value, and compares between them.

### Filter Categories

- **Boolean:** Determines whether a Cortex XSOAR field is true or false, or the string representation is true or false.

- **Date:** Determines whether the left-side time value is earlier than, later than, or the same time as the right-side time value.

Supported time and date formats:

Format	Example
ANSIC	Tues Jan _2 15:04:05 2019
UnixDate	Tues Jan _2 15:04:05 MST 2019
RubyDate	Tues Jan 02 15:04:05 -0700 2019
RFC822	02 Jan 19 15:04 MST
RFC822Z	02 Jan 19 15:04 -0700 // RFC822 with numeric zone
RFC850	Tuesday, 02-Jan-19 15:04:05 MST
RFC1123	Tues, 02 Jan 2019 15:04:05 MST
RFC1123Z	Tues, 02 Jan 2019 15:04:05 -0700 // RFC1123 with numeric zone
RFC3339	2019-01-02T15:04:05Z07:00
RFC3339Nano	2019-01-02T15:04:05.999999999Z07:00
Kitchen	3.04PM
Stamp	Jan _2 15:04:05
StampMilli	Jan _2 15:04:05.000
StampMicro	Jan _2 15:04:05.000000
StampNano	Jan _2 15:04:05.000000000

The following table shows the *After* date filter:

Left-Side Value	Right-Side Value	Result
Mon, 02 Jan 2006 15:04:05 MST	Mon, 02 Jan 2003 15:04:05 MST	True
Mon, 02 Jan 2003 15:04:05 MST	Mon, 02 Jan 2006 15:04:05 MST	False
Mon, 02 Jan 2003 15:04:05 MST	1136239445	False

- **General:** Includes general filters, such as contains, doesn't contain, In, empty, etc.

The following table shows an example of the *Contains* general filter:

Left-Side Value	Right-Side Value	Result
a, b, c	a	True
a, b, c	d	False

- **String:** Determines the relationship between the left-side string value and the right-side string value, such as starts with, includes, in list, and so on. The string filter returns partial matches as True.

The following table shows an example of the *Matches (regex)* string filter.

Left-Side Value	Right-Side Value	Result
Dbot makes your life easy	.*easy	True

- **Number:** Determines the relationship between the left-side number value and the right-side number value, such as equals, greater than, less than, etc.

The following table shows an example of the *Doesn't equal* number filter:

Left-Side Value	Right-Side Value	Result
8	2	True
8	8	False

- **Unknown:** Miscellaneous filter category

## Transformers Operators

Transformers enable you to take one value and transform or render it to another value. When you have more than one transformer, you can reorder them using click-and-drag.

Note the following:

- Transformers try to cast the transformed value (and arguments) to the necessary type. Tasks will fail if casting has failed, for example (`{ "some": "object" } To upper case => Error`)
- Some transformers are applied on each item of the result. For example, `a, b, c To upper case => A, B, C`.
- Some transformers operate on the entire list. For example, `a, b, c count => 3`.
- Some transformers are implemented as automations (meaning custom transformers automation with the **transformer** tag. You can find examples in the automation description. For more information about creating custom transformers, see [Create Custom Filters and Transformers Operators](#).

### Transformer Categories

**Date:** Transforms the date. For example:

Name	Description	Example
Date to string	Converts any date to a specified string format to the given format. String to format. By default RFC822 (02 Jan 06 15:04 MST).	<code>Mon, 02 Jan 2006 15:04:05 MST =&gt; 02 Jan 06 15:04 MST.</code>

Name	Description	Example
Date to Unix	Converts any date to Unix format. See the <a href="#">Filter Operators</a> for a list of supported time and date formats.	<b>Mon, 02 Jan 2006 15:04:05 MST =&gt; 1136214245</b>

**General:** Includes general transformers, such as sort, splice, stringify, etc. The following table describes the General examples:

Name	Description	Example
Unique	Returns a de-duped version of a list.	<b>a, b, a, c, d, a, b =&gt; a, b, c, d</b>
Slice	Returns part of a specified list in a range of <b>from</b> index (included) through <b>to</b> index (not included)  <b>from:</b> Zero based index at which to begin extraction (default: 0)  <b>to:</b> Zero based index before which to end extraction (default: list length)	<b>a, b, c, d from: 1, to: 3 =&gt; b, c</b>
Slice by item	Returns part of a list specified in a range of from item (included) through to item (not included).  <b>from:</b> Item from which to begin the extraction. If not specified, extracts from the beginning of the list.  <b>to:</b> Item before which to end the extraction. If not specified, extracts from the end of the list.	<b>a, b, c, d from: b, to: c =&gt; b, c</b>
Sort	Sorts an entire list. Supports strings and numbers. descending: <b>true</b> to sort in descending order, default is false.	<b>b, c, a =&gt; a, b, c</b> <b>2.1, 1.2, 3.4 descending: true =&gt; 3.4, 2.1, 1.2</b>
Get index	Get item at the given index. Index: Index of the item to get.	<b>b, c, a index: 0 =&gt; b</b> <b>b, c, a index -1 =&gt; nil</b>
Splice	Adds or removes items to/from an array. index: (required) Zero-based index at which to begin add/remove items. deleteCount: Number of elements to remove from 'index', default is 0. item: Item to add to the array after 'index' position.	<b>a, b, c, d, index: 1 deleteCount: 2 =&gt; a, d</b> <b>a, b, c, d, index: 2 item: w =&gt; a, w, c, d</b>
Index of	Returns the first index of the element in the array, or -1 if not found.  item: Item to locate in the array.  fromLast: <b>true</b> to get the index from last. (default is false).	<b>a, b, a, c, d, a, b, item: b =&gt; 1</b> <b>a, b, a, c, d, a, b, item: a fromLast: true =&gt; 5</b>

Name	Description	Example
		<code>a, b, a, c, d, a, b, item: w =&gt; -1</code>
Get field	Extracts a given field from the given object. field: (required) The field to extract from the result	<code>{"name": "john", "color": "white"} field: "color" =&gt; "white"</code>
Stringify	Converts the given item to a string.	<code>{ "name": "john", "color": "white" } =&gt; '{"name": "john", "color": "white"}'</code>
Count	Returns the number of elements.	<code>b, c, a =&gt; 3</code> <code>null =&gt; 0</code> <code>a =&gt; 1</code>
Join	Concatenates all elements. separator: Specifies a string to separate each pair of adjacent elements of the array, default is an empty string.	<code>b, c, a separator: , =&gt; b,c,a</code> <code>b, c, a =&gt; bca</code>

**String:** Transforms strings. To make regex case non-sensitive, use the `(?i)` prefix (for example `(?i) yourRegexText`). The following table describes string examples.

Name	Description	Example
replace match	Returns a string with some or all matches of a regex pattern, and replaces with a specified string. regex: A regex pattern to be replaced by the <code>replaceWith</code> argument. replaceWith: The string that replaces the string specified in the <code>toReplace</code> argument, default is an empty string. Detailed RegEx syntax can be found on <a href="https://github.com/google/re2/wiki/Syntax">https://github.com/google/re2/wiki/Syntax</a>	<code>pluto, is, not, a, planet</code> <code>regex: "," replaceWith: ";"</code> <code>=&gt; "pluto;is;not;a;planet"</code> <code>"pluto is not a planet" regex .*to replaceWith vega =&gt; vega is not a planet</code>
Substring	Returns a subset of a string between one index and another, or through the end of the string. from (required): An integer between 0 and the length of the string, specifying the offset into the string of the first character to include in the returned substring. to (optional): An integer between 0 and the length of the string, which specifies the offset into the string of the first character not to include in the returned substring.	<code>pluto is not a planet from: 4 to: 10 =&gt; o is n"</code>
Split	Splits a string into an array of strings, using a specified delimiter string to determine where to make each split.	<code>hello world,bye bye world =&gt; hello world, bye bye world</code> <code>hello world delimiter</code>

Name	Description	Example
	delimiter: Specifies the string which denotes the points at which each split should occur, default delimiter is ,.	=> <b>hello, world</b>
Split & trim	Splits a string into an array of strings and removes whitespace from both ends of the string, using a specified delimiter string to determine where to make each split. Arguments delimiter: Specifies the string which denotes the points at which each split should occur (default delimiter is ",").	<b>hello &amp; world</b> delimiter: & => <b>hello, world</b>
From string	Returns a subset of a string from the first from string occurrence. from (required): String to substring from.	<b>pluto is not a planet</b> from: <b>pluto is</b> => <b>not a planet</b>
To string	Returns a subset of a string until the first to string occurrence. to (required): String to substring until.	<b>pluto is not a planet</b> to: <b>a planet</b> => <b>pluto is not</b>
concat	Returns a string concatenated with given prefix and suffix. prefix: A prefix to concat to the start of the argument. suffix: A suffix to concat to the end of the argument.	<b>night</b> prefix <b>good</b> => <b>good night</b> <b>night</b> suffix <b>shift</b> => <b>night shift</b>

Number: Transforms a number. Examples:

Name	Description	Example
Floor	Returns the highest integer less than or equal to the number.	<b>1.2</b> => <b>1</b>
Ceil	Returns the lowest integer greater than or equal to the number.	<b>1.2</b> => <b>2</b>
Round	Returns the nearest integer, rounding half way from zero.	<b>7.68</b> => <b>8</b> <b>2.43</b> => <b>2</b> <b>2.5</b> => <b>3</b>
Absolute	Returns the absolute value of the given number.	<b>-2</b> => <b>2</b>
Decimal precision	Truncates the number of digits after the decimal point, according to the by argument. by: Number of digits to keep after the decimal point, default is 0.	<b>8.6666</b> by <b>2</b> => <b>8.66</b>

Name	Description	Example
Modulus (remainder)	The modular operator (%) returns the division remainder. by (required): Modulo by, default:0	<b>20 by: 3 =&gt; 2</b>
To percent	Converts a number to a percent. withsign: Specify true to include %. Default is false	<b>0.22 =&gt; 20</b> <b>0.22 withsign: true =&gt;20%</b>
Quadratic equation	Returns the result of the Quadratic Formula. b (required): The b number of: $ax^2 + bx + c = 0$ , default is 0. c (required): The c number of: $ax^2 + bx + c = 0$ , default is 0.	<b>1 b: 3 c: 2=&gt; -1.00, -2.00</b> <b>3 b: 2 c: 4=&gt; (-0.333 +1.106i), (-0.333 -1.106i)</b>

## Create Custom Filters and Transformers Operators

If you require a filter or transformer operator that is not provided out of the box, you can create your own by creating a script and then adding to the operators window.

**STEP 1** | Select **Automation > New Automation**.

**STEP 2** | Type a meaningful name for the Automation script, and click **Save**.

**STEP 3** | To create a filter operator script, do the following:

1. In the **Tags** field, add the **filter** tag.

If you want a custom transformer that operates on an entire array rather than on each individual item, you need to add the **entirelist** tag.

2. In the **Arguments** section, add the following arguments:

Argument	Description
left	Mark as mandatory. This argument defines the left-side value of the transformer operation. In this example, this is the value being checked if it falls within the range specified in the right-side value.
right	Mark as mandatory. This argument defines the right-side value of the transformer operation. In this example, this is the range to check if the left-side value is in.

### Script settings ✕

**Basic**

Name:  Type: Python ▾

Description: Checks if left-side value is in range specified in the right-side value. The right-side value uses this format: (from.to notation)  
Example: InRange left=4right=1,8 will return true

Tags: filter ✕ number ✕ ▾

---

**Arguments**

Argument *	Mandatory	Default	Sensitive	Description
<input type="text" value="left"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="Value to check in the specified range."/>
Initial value				Is array <input type="checkbox"/>
List options	<input type="text" value="comma separated optional values"/>			
<input type="text" value="right"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="Range to check against"/>

3. Add the script syntax and save.

**STEP 4 |** To create a transformer operator script do the following:

1. In the **Tags** field, add the **transformer** tag.
2. In the **Arguments** section, add the following arguments:

Argument	Description
value	Mark as mandatory. The value to transform. In this example, this is the UNIX epoch timestamp to convert to ISO format.

### Script settings

**Basic**

Name:  Type: **JavaScript**

Description:   
 Example: 1525006939 will return '2018-04-29T13:02:19.000Z'

Tags: **transformer** **date**

---

**Arguments**

Argument	Mandatory	Default	Sensitive	Description
value	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Time stamp to convert

Initial value:

Is array:

List options:

3. Add the script syntax and save.

**STEP 5** | Go to the filters and transformers window and select the operator.

### Filters & operations for occurred

1 Get  
What to get out of the data bucket (e.g. File.Type)

2

3

- Splice
- Stringify
- Uniq
- WhereFieldEquals
- Date
- Date to string
- Date to unix
- TimeStampToDate
- TimeStampToDate\_1**

To upper case (String)

---

# Common Scripts to use in Automations

Common Scripts are scripts that contain common code (functions, variables, etc.) to be used across scripts which can be embedded when writing your own Automation scripts and Integrations. The common scripts appear in the Automation page, but are used to enhance the API in other scripts and integrations.

For example, the function `tableToMarkdown` in `CommonServer.yml` takes a JSON and transforms it to markdown. You can call this function from integrations and scripts that you author.

## CommonServer

`CommonServer` is a system script and cannot be changed. You can add your own functions to the `CommonUserServer` script which is a built-in script that can be modified.

You can also use `CommonUserServer` to override our existing scripts in `CommonServer`.

## Common Scripts for Python

To find common scripts for Python open the **Automation** screen and search for **CommonServerPython**.

If you have your own common scripts in Python that you would like to save and reuse you can save them in **CommonServerUserPython**.

## Common Scripts for Java Scripts

To find common scripts for Java Script open the **Automation** screen and search for **CommonServer**.

If you have your own common scripts in Java Script that you would like to save and reuse you can save them in **CommonServerUser**.

## Python Common Scripts

Script	Description	Parameter
<code>positiveUrl</code>	Gets the Entry parameter from the WarRoom and checks each Reputation Tool to determine if the URL in this Entry parameter is malicious or non-malicious.  You can change the threshold by changing the Thresholds dictionary.  The function returns true if the URL is safe, otherwise returns false.	Entry parameter
<code>positiveFile</code>	Gets the Entry parameter from the WarRoom and checks each Reputation Tool to determine if the file in this Entry parameter is malicious or non-malicious.  You can change the threshold by changing the Thresholds dictionary.  The function returns true if the file is safe, otherwise returns false.	Entry parameter
<code>vtCountPositives</code>	Gets the entry parameter and checks how many good URLs are hosted on the IP in this entry.	Entry parameter
<code>shortCrowdStrike</code>	Formats the response from CrowdStrike to a pretty markdown.	Entry parameter

Script	Description	Parameter
shortUrl	Gets the Entry parameter from the War Room and checks it in the Reputation tools (adding information to context and formatting the response to the War Room) when checking the URL.	Entry parameter
shortFile	Gets the Entry parameter from the War Room and checks it in the Reputation tools (adding information to context and formatting the response to the War Room) when checking the IP.	Entry parameter
FormatADTimestamp	Gets the Entry parameter from the War Room and formats the timestamp returned from AD.	Entry parameter
formatCell	Gets the JSON parameter string and formats it to a regular string that can be used in a table.	JSON parameter string
flattenCell	Gets the JSON parameter string and converts it to a string that can be used in a table. Also supports tables containing sub cells.	JSON parameter string
flattenRow	Gets a key and data and adds it to the context. Checks if the key already exists. If it exists, it creates an array in the existing key.	Key and data, and optionally dedup=False – don't add duplicate items
fileResult	Creates a new file that contains the data and displays the file in the WarRoom.	Filename and data

### JavaScript Common Scripts

Script	Description	Parameter
tableToMarkdown	Converts a Cortex XSOAR table in JSON format to a Markdown format table.	Table name, JSON object and the headers to display.
closeInvestigation	Closes the current investigation.	args - arguments for the close (what happened, damage, etc.)
setSeverity	Sets the severity of an incident. The incident must be related to the current investigation.	arg that has 2 keys: <ul style="list-style-type: none"> <li>'id' - the incident id</li> <li>'severity' - the new severity value (Critical, High, Medium etc.)</li> </ul>
setIncident	Sets fields of the incident. The incident must be related to the current investigation and be the only incident in it.	Dictionary of args - has 5 optional keys: <ul style="list-style-type: none"> <li>type</li> <li>severity</li> </ul>

Script	Description	Parameter
		<ul style="list-style-type: none"> <li>• details</li> <li>• name</li> <li>• incident systems</li> </ul>
createNewIncident	Creates a new incident with the fields specified. This is only carried out if an incident with the same name does not exist as an active incident.	Dictionary of args - has 5 optional keys: type, severity, details, name and the incident systems.
setOwner	Sets the owner of the incident. The incident must be related to the current investigation.	Owner user name.
taskAssign	Assigns a playbook task to a user.	Dictionary of args - has 2 keys: <ul style="list-style-type: none"> <li>• 'id' - the task id</li> <li>• 'assignee' - assignee user name</li> </ul>
setPlaybook	Sets investigation playbook	Playbook name.
isCommandAvailable	Checks if the specific command is available.	Command name.
objToMd	Converts a given object to Markdown while descending multiple levels.  Gets the object that will be converted to Markdown.	Object
positiveUrl	Gets the Entry parameter from the WarRoom and checks each Reputation Tool if the URL in this Entry parameter is malicious or non-malicious.  You can change the threshold by changing the Thresholds dictionary.  The function returns true if the URL is safe, otherwise returns false.	Entry from War Room.
positiveFile	Gets the Entry parameter from the WarRoom and checks each Reputation Tool if the file in this Entry parameter is malicious or non-malicious.  You can change the threshold by changing the Thresholds dictionary.  The function returns true if the file is safe, otherwise returns false.	Entry from War Room.
positiveIP	Gets the Entry parameter from the WarRoom and checks each Reputation Tool if the IP in this Entry parameter is malicious or non-malicious.	Entry from War Room.

Script	Description	Parameter
	<p>You can change the threshold by changing the Thresholds dictionary.</p> <p>The function returns true if the IP is safe, otherwise returns false.</p>	
shortCrowdStrike	Formats the response from CrowdStrike to a pretty Markdown.	Entry from War Room.
shortUrl	Gets the Entry parameter from the War Room and checks it in the Reputation tools (adding information to context and formatting the response to the War Room) when checking the URL.	Entry from War Room.
shortFile	Gets the Entry parameter from the War Room and checks it in the Reputation tools (adding information to context and formatting the response to the War Room) when checking the file.	Entry from War Room.
shortIp	Gets the Entry parameter from the War Room and checks it in the Reputation tools (adding information to context and formatting the response to the War Room) when checking the IP.	Entry from War Room.
treeToFlattenObject	Flattens all JSON tree objects to key-value format.	JSON object.



# *Work with SLAs*

- > SLA Overview
- > Create an SLA Field
- > Manage SLA and Timer Fields in an Incident
- > Create an SLA Trigger
- > Customize SLA Scripts
- > Search Incidents using SLA and Timer Fields
- > Configure the Global Risk Threshold



# SLA Overview

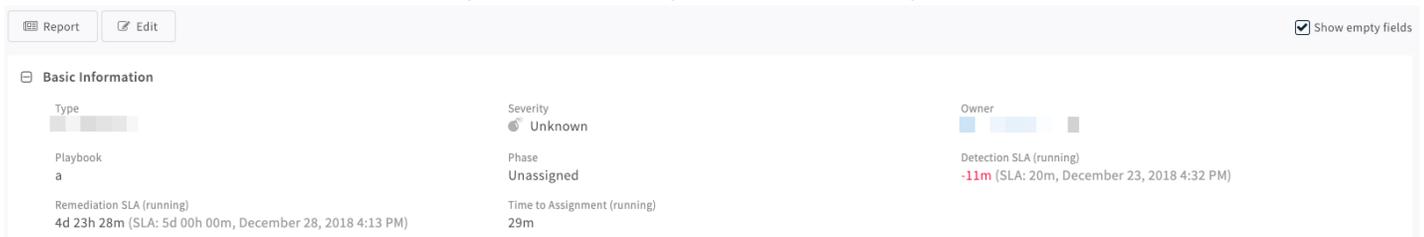
Cortex XSOAR supports specific fields for managing SLAs and timers.

SLAs are an important aspect of case management. You can incorporate SLA fields in your cases so you can view how much time is left before the SLA becomes past due, as well as configure actions to take in the event that the SLA *does* pass.

In addition, you can now view the number of cases that are at risk of passing the SLA, or are already late, using pre-configured widgets. The widgets present information based on the default threshold, [which can be configured globally](#).

## Present SLAs in Incident Summary Layouts

Once you have configured the SLA fields and timers, your incident summary screens will display information about the status of the SLA, if any of the SLAs are past due, and if so, by how much.



In the image above, for example, we see that the timers for several of the fields are in various states. **Detection SLA** is past due, while **Remediation SLA** has nearly 5 days remaining.

## Customize CSV Reports for SLA Fields

You can add SLA specific information to your CSV reports. Edit the table columns field in the JSON report to include the SLA data that you want.

For example, assuming that you have an existing timer field named `myslatimer`, we can use the following options as csv columns:

- `myslatimer`: displays a summary of the timer status and sla.
- `myslatimer.runStatus`: displays a run status of the current timer.
- `myslatimer.totalElapsed`: displays the total elapsed time, in seconds, of the current timer. If the timer has ended, it displays the total duration.

# Create an SLA Field

By default, the system comes with several SLA and Timer fields.

The SLA fields count down the time remaining, while as a timer field it serves as a counter, counting the time that has elapsed since started. In the event that no SLA is defined, the SLA fields serve as a counter.

**STEP 1** | Navigate to **Settings > Advanced > Fields**.

**STEP 2** | Click **+New Field**.

**STEP 3** | Under **Field Type** field, select **Timer/SLA**.

**STEP 4** | Enter a name and optional tooltip for the field.

**STEP 5** | Define a duration for the SLA of this field. If no value is entered, the field serves as a counter.

**STEP 6** | Determine the risk threshold for this timer. When the timer falls below this threshold, it is considered at risk. By default, the threshold is 3 days, [which is defined in the global system parameter](#).

**STEP 7** | Under **Run on SLA Breach**, select the script to run when the SLA time has passed. For example, email the supervisor or change the assignee.

Only scripts to which you have added the SLA tag will appear in list of scripts that you can select.

The screenshot shows the 'New Field' configuration form. At the top, the title is 'New Field' with a close button (x). The form is divided into two tabs: 'Basic Settings' (selected) and 'Attributes'. Under 'Basic Settings', there is a 'Field Type' dropdown menu set to 'Timer/SLA'. Below that is the 'Field Name' field with the value 'TimeToNotify' and a tooltip field with the value 'Informs you of the stage of the investigation'. The 'Machine name' is 'timetonotify' with a note '(use in search and command line)'. The 'SLA' section has a value of '3' days and '00' hours. The 'Risk Threshold' section has a value of '1' day and '00' hours. The 'Run on SLA Breach' dropdown is set to 'SendEmailOnSLABreach'. At the bottom right, there are 'Cancel' and 'Save' buttons.

---

STEP 8 | Click **Save**.

---

# Manage SLA and Timer Fields in an Incident

You can manage the timers and SLA for a specific incident. This enables you to manage SLAs on a global level within the SLA fields, and on a more granular level within specific incidents when the need arises. For example, if the severity of the incident dictates that you decrease the response time for the given incident.

## STEP 1 | SLA Fields

You can use commands to set the SLA for a specific SLA field in a specific incident using the `setIncident` command and adding the SLA field for which to set the time.

If you do not enter a value for the `slaField` parameter, the time you enter is applied to the incident's Due Date.

### Example

The following example shows you how to change the Time to Detection field to 30 minutes for the current incident:

```
!setIncident sla=30 slaField=timetodetection
```



*When defining the values for the `slaField` and timer commands, all values must be in lowercase and cannot have any spaces.*

## STEP 2 | Timer Fields

You can use commands to change the state of a timer for an incident using the following:

- `startTimer` - Starts the timer. This command should also be used to restart a paused timer.



*Timers are not started automatically when an incident is created.*

- `pauseTimer` - Pauses the timer.
- `stopTimer` - Stops the timer. Once a timer is stopped, you can only reset a timer using the `resetTimer` command.



*Timers are automatically stopped when an incident is closed.*

- `resetTimer` - Resets a timer. This command should be used to enable a timer that was stopped.

### Example

The following example shows you how to pause a timer for a specific field in the current incident:

```
!pauseTimer timerField=timetodetection
```

You can specify the `incidentID` to change the timer for a different incident.

---

# Create an SLA Trigger

Timers are specific tasks that you implement in playbooks to manage SLAs. The timers can be triggered to start, pause, or stop when a certain task occurs. For example, a timer can be triggered to stop for the **Time to Assign** field when the incident is assigned an owner, and to immediately start the timer for the **Time to Remediation** field.

**STEP 1** | Navigate to the **Playbooks** page.

**STEP 2** | Select the playbook to which you want to add the timer and click **Edit**.

**STEP 3** | Click the + symbol to add a new task or click an existing task to edit the task.

**STEP 4** | In the Timers tab, select the action that you want the timer to perform for the given task. Valid options are:

- Start - Starts the timer.



*Timers are not started automatically when an incident is created.*

- Pause - Pauses the timer.
- Stop - Stops the timer. Once a timer is stopped, you can only reset a timer using the `resetTimer` command.



*Timers are automatically stopped when an incident is closed.*

**STEP 5** | Select the field on which the timer is applied.

**STEP 6** | Click **Ok**.

# Customize SLA Scripts

Scripts in Cortex XSOAR enable you to automate processes. In the context of SLA, you can create scripts that will perform specific actions when the SLA is breached. Each SLA script must include the SLA tag.

Cortex XSOAR comes with an out-of-the-box script, called `SendEmailOnSLABreach`, that sends an email to specific users when the script is triggered. By default, the script sends an email to the incident assignee, but you can configure additional recipients within the script.

When you create your own scripts, the following arguments are automatically added, in addition to the basic elements provided with every script (for example, current investigation and current incident):

- `field` - the current triggered sla breach field object (contains: name, cliName, threshold, etc)
- `fieldValue` - the current triggered sla field's value, for example the `startDate`.

The following table lists the different properties in the SLA timer field value:

Property	Type	Description
<code>dueDate</code>	Date	The date by which the SLA for this timer is due.
<code>breachTriggered</code>	Boolean	Was the timer already in breach of the SLA.
<code>sla</code>	INT (in minutes)	The period defined as the SLA for this timer. This is the value that you defined in the timer field.
<code>endDate</code>	Date	The date at which the SLA timer completed.
<code>lastPauseDate</code>	Date	The last date at which the SLA timer was paused.
<code>startDate</code>	Date	The date at which the SLA timer was started.
<code>accumulatedPause</code>	INT (in seconds)	The total number of seconds that the timer was in a paused state.
<code>totalDuration</code>	INT (in seconds)	The total number of seconds that the timer was running. This property is populated after the timer is stopped.
<code>slaStatus</code>	INT	Represents the Demisto SLA status. Values are: 0 - The SLA is within the allotted range. 1 - The SLA is below the defined risk threshold. 2 - The SLA is in breach.
<code>runStatus</code>	String	Represents the current status of the timer. Values are: idle running paused ended

---

# Search Incidents using SLA and Timer Fields

You can search for incidents based on their SLA in several ways:

- based on the SLA status



*The SLA status is not defined unless the timer is in a stopped mode, meaning, either paused or ended.*

- based on an SLA field
- based on a timer field

For example, you can search for all of the timer fields that are currently running, or you can search for all incidents with a specific SLA status.

**STEP 1 |** Navigate to the **Incidents** page.

**STEP 2 |** To search for an incident whose timer is still active, enter the following:

- The name of the field
- The run status
- The due date. This is required for queries who run status is neither ended nor paused to improve query performance.

**STEP 3 |** To search for an incident whose timer is no longer active, enter the SLA Status.

## Examples

In the following example, we are searching for all incidents that have an SLA timer called slatimer and fulfill the following criteria:

- The run status is neither ended nor paused AND the due date is later than now, meaning, the due date has not yet passed.

OR

- Incidents whose run status is ended or paused and the SLA status is within the allotted time.

```
(-slatimer.runStatus:(ended paused) and slatimer.dueDate:>"now") or
(slatimer.slaStatus:"within")
```

In the following example, we are searching for all incidents that fulfill the following criteria:

- The run status is either ended or paused AND the due date is earlier than now, meaning, the due date has already passed.

OR

- Incidents whose run status is ended or paused and the SLA status is late.

```
(-slatimer.runStatus:(ended paused) and slatimer.dueDate:<"now") or
(slatimer.slaStatus:"late")
```

In the following example, we are searching for all incidents that fulfill the following criteria:

- The run status is neither ended nor paused AND the due date is between now and 5 hours. The 5 hours represents our risk threshold.

---

OR

- Incidents whose run status is ended or paused and the SLA status is Risk.

```
(-slatest.runStatus:(ended paused) and slatest.dueDate:>"now" and
slatest.dueDate:<"in 300 minutes") or (slatest.slaStatus:"risk")
```

---

# Configure the Global Risk Threshold

By default, the risk threshold is 72 hours. You can change the threshold by adding a parameter to the system settings.

**STEP 1** | Navigate to **Settings > About > Troubleshooting**.

**STEP 2** | Under Server Configuration, click **Add Server Configuration**.

1. In the key field, enter `sla.risk.threshold`.
2. In the value field, enter the number, in hours, to which to set the risk threshold.

**STEP 3** | Click **Save**.



# *Machine Learning Models*

- > Machine Learning Models Overview
- > Create a Machine Learning Model
- > Phishing Classifier Demo



---

# Machine Learning Models Overview

Machine learning models enable Cortex XSOAR to analyze and predict behavior through incident types and fields. The model uses past incidents that have already been classified to classify incoming events automatically.

Machine learning models are used mainly for phishing incidents. You can train it to automatically recognize, for example, phishing emails, emails that are legitimate, and those that contain spam.

Machine learning models enable you to do the following:

- Use as part of a scoring/severity set.
- To close incidents automatically more accurately than manually defining a threshold.
- Handle only incidents that the classifier marks as malicious.

You train models by inputting data through incident types and fields. Cortex XSOAR returns all the incidents containing the specified field. You can then map these field values into different verdicts. The verdicts determine what the model predicts, so you should make the verdict definitions meaningful.

By default, Cortex XSOAR trains models from input data contained in an Email body, Email HTML, and Email subject. You can change the name of the fields containing the subject and body. Cortex XSOAR then trains a model and returns the accuracy of the model against each category.

To create a machine learning model, see [Create a Machine Learning Model](#).

The machine learning model for phishing can be used as following:

- Part of the **Phishing Investigation - Generic v2** playbook, when adding the **DbotPredictPhishingWords** command, or when creating a playbook.

When Cortex XSOAR runs the playbook it takes the machine learning model that you have defined.

- Run the **!DbotPredictPhishingWords** command in the War Room or in the Machine Learning page, by typing: **!DbotPredictPhishingWords modelName="name" emailBody="body"emailbodyhtml="email body html" emailsubject="email subject"**. For examples, see [Phishing Command Examples Using a Machine Learning Model](#)

You can run a [phishing classifier demo](#), without the need to create a machine learning model.

---

# Create a Machine Learning Model

A machine learning model enables Cortex XSOAR to predict the classification of phishing incidents. For example whether the incident should be classified as legitimate, malicious, or spam. You can use these models in conjunction with your default investigation playbooks or run commands separately in the War Room. It is usually used for training a model to predict the classification of a phishing incident. The main goal of the machine learning model is leveraging past phishing incidents to assist with the investigation of future incidents.

**STEP 1** | Select **Settings > Advanced > ML Models > New Model**.

**STEP 2** | In the **Model name** field, type the name of the model that you want to create.

**STEP 3** | (Optional). In the **Description** field, type a meaningful description for the machine learning model.

**STEP 4** | In the **Incident type** field, from the drop down list, select the type of incident where you want you want to the machine to be trained, such as Phishing.

**STEP 5** | In the **Incident field**, from the drop down list, select the incident field where you want the model to learn to predict. The model trains using these fields as a label. For example, **Email Classification**.

**STEP 6** | Select the date range where you want to run the machine learning. The more incidents, the better results. It is recommended to use a longer period.

**STEP 7** | In the **Maximum number of incidents to test**, type the number you want to test that is used to train the model. Reduce the number only if the number of incidents is too large and causes performance problems. Use a higher number if you have more samples in your environment Default is 3000. The results appear in the **Field Mapping** field.

**STEP 8** | In the **Verdict** fields, define the name of the verdict for which to map your data.

Verdicts are group of labels, for which each verdict includes 1 label or more. You must map all existing labels into 2 or 3 different verdicts. The model is trained using these verdicts. All labels that are mapped into the same verdict are treated as if they have the same label. You can choose any label for your verdict field, but the training model calculates the model based on the verdict, so it should be a meaningful name.

**STEP 9** | In the **Field Mapping** field, drag and drop the **Field Mapping** data into **Verdict** fields.

You need a minimum of 50 results returned. For an example. see [Machine Learning Model Example](#).

**STEP 10** | If you want to change the fields where email body and email subject are stored in the incident, in the **Argument Mapping** select the equivalent fields for Email body, Email HTML and Email subject.

By default, the machine learning model trains the Email body, Email HTML and Email subject.

**STEP 11** | Click **Start Training**.

In the **ML Models** window, the machine learning model starts analyzing the data. When finished, if successful, the percentage scores appear, which reflect how precise the results are according to class. If

using the phishing incident type, you can now use model in the machine learning or War Room window or in the playbook. For more information, see [Machine Learning Models Overview](#).

## Machine Learning Model Example

In this example, we want to create a machine learning model for **Phishing** using a **customlabelfield**. The **customlabelfield** manually classifies data as to Phishing, Spam and legit.

1. From the **New ML Model** window, in the **Incident type** field, select **Phishing**.
2. In the **Incident field**, select the field which contains the label you want the model to train. Assume that the field which contains the type of the mail is called **customlabelfield**.

In the **Field Mapping** field, the following data is returned:

### New ML Model - Phishing machine learning classifier

The ML model trains on incident field and the field values are mapped to a model verdict.

Model name\*

Description

Incident type\* **Phishing** ▼

Date range 

Maximum number of incidents to test\*

Incident field\* **Customlabelfield** ▼

Drag field values to the appropriate DBot Verdict column (max. 3 DBot Verdict columns)

Field Mapping	Verdict: ▼	Verdict: ▼	Verdict: ▼
<b>legit (2023)</b> <b>phishing (185)</b> <b>spam (792)</b>			
	Total:	Total:	Total:

3. Define the Verdict fields according to malicious, spam, and legit.

Field Mapping	Verdict: <b>Malicious *</b> ▼	Verdict: <b>Spam *</b> ▼	Verdict: <b>Legit *</b> ▼
<b>legit (2023)</b> <b>phishing (185)</b> <b>spam (792)</b>			
	Total: 0	Total: 0	Total: 0

4. Drag and drop the data from **Field Mapping** into the relevant **Verdict** fields.

Field Mapping	Verdict: Malicious *	Verdict: Spam *	Verdict: Legit *
	phishing (185)	spam (792)	legit (2023)
	Total: 185	Total: 792	Total: 2023

- In the **Argument mapping** field keep the arguments as the default. We want the machine learning model to train on Email body, HTML and Subject.
- Click **Start Training**.

The model starts training and when successful returns the following information:

demoModel legit 96% phishing 81% spam 97%

Incidents: 483 Date range: 25 Jul 2019 - Incident type: Phishing Field: customlabelfield  
Description: model for demo

The returned data shows that it found 3 categories together with the percentage scores, which reflect the precision of the results.

You can now use the machine learning model in the **Phishing Investigation - Generic V2** playbook, in the Machine Learning page or in the War Room. For examples how to use it in the War Room, see [Phishing Command Examples Using a Machine Learning Model](#).

## Phishing Command Examples Using a Machine Learning Model

In this example, we have created a machine learning model, called “demoModel” that predicts the following:

demoModel legit 96% phishing 81% spam 97%

Incidents: 483 Date range: 25 Jul 2019 - Incident type: Phishing Field: customlabelfield  
Description: model for demo

For an example how to create the machine learning model, see [Machine Learning Model Example](#).

After running the command, Cortex XSOAR returns the following information:

- **TextTokensHighlighted:** The text of the email message with the highlighted positive words (if found).
- **Label:** The predicted label found by the model.
- **Probability:** The prediction probability.
- **PositiveWords:** Words that encouraged the model to make the prediction.
- **NegativeWords:** Words that are in general not correlated with the predicted class and reduced the model’s confidence in its prediction.

In the War Room, run the following commands:

```
!DBotPredictPhishingWords modelName="demoModel" emailBody="Your email account was LOGIN today by Unknown IP address: 10.240.180.228, click on UPDATE"
```

<<http://helpd.moonfruit.com/>> to validate and verify your email account now to avoid Outlook Web App been disabled for user"

DBot  
January 21, 2020 1:27 PM

Command: `IDBotPredictPhishingWords modelName="demoModel" emailBody="your email accou...` (Scripts)

DBot Predict Phishing Words

TextTokensHighlighted	your email account was login today by unknown ip address: 10.240.180.228, click on update < <a href="http://helpd.moonfruit.com/">http://helpd.moonfruit.com/</a> > to validate and verify your e-mail account now to avoid outlook web app been disabled for user
Label	phishing
Probability	1.00
PositiveWords	App, verify, account, Outlook, email
NegativeWords	NUMBERPATTERN, address

`!DBotPredictPhishingWords modelName="demoModel" emailBody="Your Outlook Exceeded its storage limit Click here <https://docs.google.com/forms/d/e/1FAIpQLSckF75SUGErVfMTEfHhhFkiX2-4V2tgC0nssDvpkqZnPz4pkQ/viewform> fill and SUBMIT for more space or you wont be able to send Mail."`

DBot  
January 26, 2020 3:01 PM

Command: `IDBotPredictPhishingWords modelName="demoModel" emailBody="our Outlook excee...` (Scripts)

DBot Predict Phishing Words

TextTokensHighlighted	our outlook exceeded it's storage limit. click here < <a href="https://google.com/form/d/e/1/faipqlsckf75sugervfmtafhfh-fkix2-4v2tgc0nssdvpkqznpz4pkq/viewform">https://google.com/form/d/e/1/faipqlsckf75sugervfmtafhfh-fkix2-4v2tgc0nssdvpkqznpz4pkq/viewform</a> > fill and submit for more space or you won't be able to send mail.
Label	phishing
Probability	0.95
PositiveWords	Outlook, space, able, storage, exceed, SUBMIT, limit, Mail
NegativeWords	will, Click, send

`!DBotPredictPhishingWords modelName="demoModel" emailBody="Dear member, the credit card we have on file for your PayPal service was declined when we attempted to bill you for your most recent service fees. For this reason, your service could be suspended. You must update your billing information immediately in order to avoid any interruption to your services"`

DBot  
January 26, 2020 3:58 PM

Command: `IDBotPredictPhishingWords modelName="demoModel" emailBody="Dear member, the credit card we have on f...` (Scripts)

DBot Predict Phishing Words

TextTokensHighlighted	**** dear member, the credit card we have on file for your paypal service was declined when we attempted to bill you for your most recent service fees. for this reason, your service could be suspended. you must update your billing information immediately, in order to avoid any interruption to your services
Label	phishing
Probability	0.99
PositiveWords	Dear, immediately, update, file
NegativeWords	credit

`DBotPredictPhishingWords modelName="demoModel" emailBody="lose 22.5lbs in 3 weeks! flush fat away forever! free 30-day supply **http://www.adclick.ws/p.cfm?o=423&s=pk19** to unsubscribe, click below: http://u2.azoogle.com/?z=93-1090346-6211c4"`

 **DBot**  
January 26, 2020 3:52 PM

Command: `IDBotPredictPhishingWords modelName="demoModel" emailBody="lose 22.5lbs in 3 w..."` (Scripts)

**DBot Predict Phishing Words**

TextTokensHighlighted	lose 22.5lbs in 3 weeks! flush fat away forever! free 30-day supply **** <a href="http://www.adclick.ws/p.cfm?o=423&amp;s=pk19.****">http://www.adclick.ws/p.cfm?o=423&amp;s=pk19.****</a> to unsubscribe, click below: <a href="http://u2.azoogle.com/?z=93-1090346-6211c4">http://u2.azoogle.com/?z=93-1090346-6211c4</a>
Label	spam
Probability	0.99
PositiveWords	week, click, Free, lose
NegativeWords	URLPATTERN, supply

---

# Phishing Classifier Demo

Cortex XSOAR comes out of the box with a pre-trained phishing classifier. The phishing classifier uses the **DBotPredictOutOfTheBox** automation, which enables you to get a prediction for a phishing incident using Cortex XSOAR's pre-trained model. After running the feature, you can see how it works in practice and then create your own machine learning models.

 *The main purpose is to demonstrate how the phishing classifier feature works, so that you learn how to train a classifier using your own data. We do not recommend using it for production.*

To run the phishing classifier, in the War Room, type **!DBotPredictOutOfTheBox**, and add the relevant parameters.

 *The **output** parameters are the same as the output of **DBotPredictPhishingWord**. The **DBotPredictPhishingWord** automation allows you to get a prediction for a phishing incident, using a model trained using your own classifier. For more information, see [Machine Learning Models Overview](#).*

For practical examples, see [DbotPredictOutOfTheBox Examples](#).

## DbotPredictOutOfTheBox Parameters

The following table describes **DbotPredictOutOfTheBox** parameters.

Parameter	Description
<b>emailBody</b>	The plain text of the email body for which you want to get the prediction.
<b>emailBodyHTML</b>	The HTML of the email for which you want to get the prediction. If the email body is filled, this field can be left empty.
<b>emailSubject</b>	The plain text of the email subject for which you want to the prediction.
<b>labelProbabilityThreshold</b>	All predictions are given in a confidence value between 0-1. If this parameter is set to 0, all model predictions are given. If more than 0, only confidence predictions higher than this value are given.
<b>minTextLength</b>	Minimum length of text (subject and body) required for getting a prediction.
<b>topWordsLimit</b>	Maximum number of words to highlight in the result.
<b>wordThreshold</b>	the lower this value is, the more words will be highlighted in the results.

## DbotPredictOutOfTheBox Parameters

The following table describes **DbotPredictOutOfTheBox** parameters.

Parameter	Description
<b>TextTokensHighlighted</b>	The text of the email message with the highlighted positive words (if found).
<b>Label</b>	The predicted label of the message.
<b>Probability</b>	The prediction probability between 0-1. The higher this value, the more confident the classifier is in its prediction.
<b>PositiveWords</b>	Words that encouraged the model to make the prediction.
<b>NegativeWords</b>	Words that are in general not correlated with the predicted class and reduced the model's confidence in its prediction.

## DbotPredictOutOfTheBox Examples

The following examples describe the parameters and output of the **DbotPredictOutOfTheBox** automation.

Run the following command in the War Room:

```
!DbotPredictOutOfTheBox emailBody="<Message>"
```

Label	Message
Malicious	Your email account was LOGIN today by Unknown IP address: 10.240.180.228, click on UPDATE <http://helpd.moonfruit.com/> to validate and verify your email account now to avoid Outlook Web App been disabled for user

DBot  
March 18, 2020 3:44 PM

Command: `!DbotPredictOutOfTheBox emailBody="Your email account was LOGIN toda..."` (Scripts)

DBot Predict Phishing Words

TextTokensHighlighted	your email account was login today by unknown ip address: 10.240.180.228, click on update <http://helpd.moonfruit.com/> to validate and verify your email account now to avoid outlook web app been disabled for user
Label	malicious
Probability	1.00
PositiveWords	validate, Web, Outlook, App, verify, click, email, account, avoid
NegativeWords	address, today

Spam	Your Outlook Exceeded its storage limit Click here <https://docs.google.com/forms/d/e/1FAIpQLSckF75SUgErVFmTEfHhhFkiX2-4V2tgC0nssDvpkqZnPz4pkQ/viewform> fill and SUBMIT for more space or you wont be able to send Mail.
------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Label	Message
-------	---------

DBot  
March 18, 2020 3:50 PM

Command: *IDBotPredictOutOfTheBox emailBody="our Outlook Exceeded its storage lim... (Scripts)*

**DBot Predict Phishing Words**

TextTokensHighlighted	our outlook exceeded its storage limit click here < <a href="https://docs.google.com/forms/d/e/1faipqlsckf75sugeryvfmetefhfhfkix2-4v2tgc0nssdvpkqznpz4pkq/viewform">https://docs.google.com/forms/d/e/1faipqlsckf75sugeryvfmetefhfhfkix2-4v2tgc0nssdvpkqznpz4pkq/viewform</a> > fill and submit for more space or you wont be able to send mail.
Label	spam
Probability	0.87
PositiveWords	wont, Click, fill
NegativeWords	Outlook, space, storage, limit, send, Mail

Malicious	Your email password expires in 2 days to retain email password and details. <b>CLICK HERE</b> <a href="https://docs.google.com/forms/d/e/1FAIpQLSewQbYraWXtr4atKnGGyNncumJfky-En54dvjVK6-Mxlu5G-A/viewform">https://docs.google.com/forms/d/e/1FAIpQLSewQbYraWXtr4atKnGGyNncumJfky-En54dvjVK6-Mxlu5G-A/viewform</a> to update immediately
-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

DBot  
March 18, 2020 3:57 PM

Command: *IDBotPredictOutOfTheBox emailBody="Your email password expires in 2 da... (Scripts)*

**DBot Predict Phishing Words**

TextTokensHighlighted	your email password expires in 2 days to retain email password and details. click here <a href="https://docs.google.com/forms/d/e/1faipqlsewqbyrawxtr4atknngynncumjfy-en54dvjvk6-mxlu5g-a/viewform">https://docs.google.com/forms/d/e/1faipqlsewqbyrawxtr4atknngynncumjfy-en54dvjvk6-mxlu5g-a/viewform</a> to update immediately
Label	malicious
Probability	0.96
PositiveWords	update, expires, details, retain, immediately, password, email
NegativeWords	days

Spam	lose 22.5lbs in 3 weeks! flush fat away forever! free 30-day supply <b>**http://www.adclick.ws/p.cfm?o=423&amp;s=pk19.**</b> to unsubscribe, click below: <a href="http://u2.azoogole.com/?z=93-1090346-62llc4">http://u2.azoogole.com/?z=93-1090346-62llc4</a>
------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

DBot  
March 18, 2020 3:59 PM

Command: *IDBotPredictOutOfTheBox emailBody="ose 22.5lbs in 3 weeks! flush fat awa... (Scripts)*

**DBot Predict Phishing Words**

TextTokensHighlighted	ose 22.5lbs in 3 weeks! flush fat away forever! free 30-day supply <b>****http://www.adclick.ws/p.cfm?o=423&amp;s=pk19.****</b> to unsubscribe, click below: <a href="http://u2.azoogole.com/?z=93-1090346-62llc4">http://u2.azoogole.com/?z=93-1090346-62llc4</a>
Label	spam
Probability	1.00
PositiveWords	forever, away, fat, click
NegativeWords	supply



# Lists

- > Work With Lists
- > Set the List Separator Character



---

# Work With Lists

A list is a collection of one or more items of the same type, for example plain text, JSON, or HTML, that you can use in scripts, playbooks, or any other place where the context button appears (double-curly brackets).

## Use cases

These are some common use cases for creating and using lists in Cortex XSOAR.

- A list of allowed executable files against which to check potentially malicious executable files.
- An HTML template that you can define to use as part of a Communication task.
- Store data object, for example JSON, that you can call as inputs for scripts and playbooks.
- Use the `getList` or `addToList` commands in a script to take action based on the list data, for example, `res = demisto.executeCommand("getList", {"listName": demisto.args()["listName"]})` will return all list entries in the script.

## List commands

You can use the following list commands in scripts and playbook tasks.

### **`getList`**

Retrieves the contents of the specified list. The command has the following required arguments.

- *listName*: the name of the list for which to retrieve the contents.

### **`createList`**

Creates a list with the supplied data. The command has the following required arguments.

- *listName*: the name of the list to which to append items.
- *listData*: the data to add to the new list.

### **`addToList`**

Appends the supplied items to the specified list. If you add multiple items, make sure you use the same list separator that the list currently uses, for example a comma or a semicolon. The command has the following required arguments.

- *listName*: the name of the list to which to append items.
- *listData*: the data to add to the specified list. The data will be appended to the existing data in the list.

### **`setList`**

Adds the supplied data to the specified list and overwrites existing list data. The command has the following required arguments.

- *listName*: the name of the list to which to append items.
- *listData*: the data to add to the specified list. The data will overwrite the existing data in the list.

### **`removeFromList`**

Removes a single item from the specified list. The command has the following required arguments.

- *listName*: the name of the list from which to remove an item.
- *listData*: the item to remove from the specified list.

---

## Create a List

Predefined lists can be created and modified to be used in scripts and War Rooms. A list can contain items of the same type in any format that would be useful for you. These are later parsed by scripts, and can be modified by scripts. For example, you might need to create a list of e-mails, a list of known trusted IPs (allow list), JSON objects, etc.

Cortex XSOAR includes the following out of the box lists:

- Indicators exclusion list
- Internal IP ranges

**STEP 1** | Go to **Settings** > **Advanced** > **Lists**.

**STEP 2** | Click **Add a List**.

**STEP 3** | Type a name for the list.

**STEP 4** | Add the required data.

**STEP 5** | Click **Save**.

**STEP 6** | (Optional) To modify the list, select the list and click the edit icon.

---

# Set the List Separator Character

You can set a custom separator globally (all lists) or for individual lists. The custom list separator can only be a single character, for example a single semicolon (;) or a single colon (:). If you set the separator as multi-character, the default separator will be applied. The default separator is a comma (,).

If you set a global list separator and custom separator for an individual list, the separator set for the individual list overrides the global list separator.

**STEP 1** | Go to **Settings > About > Troubleshooting**.

**STEP 2** | In the **Server Configuration** section, click **Add Server Configuration**.

**STEP 3** | Add the necessary key and value.

Type	Key	Value
Global	<code>list.separator</code>	Single-character separator. For example “;”.
List	<code>list.&lt;listName&gt;.separator</code>	Single-character separator. For example “;”.



# ***Cortex XSOAR Enterprise Mobile App***

- > [Cortex XSOAR Enterprise Mobile App Overview](#)
- > [Android Certificate Requirements](#)
- > [Use the Cortex XSOAR Enterprise Mobile App](#)



---

# Cortex XSOAR Enterprise Mobile App Overview

The Cortex XSOAR Enterprise mobile app provides a set of features that enable you to make incident and task-based decisions on a mobile device.

The App sends the same notifications as the web application. It enables you to do the following:

- View system dashboards and incidents
- Assign analysts to incidents
- View, send and receive messages, upload, download files, and attachments in the War Room.
- Update incident types and severity
- Modify incident tasks
- Close incidents
- View, assign and mark as complete your tasks

## System Requirements

System	Minimum Requirements
Cortex XSOAR	Version 4.5 and above.
Android	<ul style="list-style-type: none"><li>• Version 4.1 (Jelly Bean, API 16) and above.</li><li>• Use a CA-Signed SSL certificate or customize the android apk. For more information, see <a href="#">Android Certificate Requirements</a>.</li></ul>
iOS	<ul style="list-style-type: none"><li>• Version 9.0 and above.</li><li>• Use a CA-signed certificate. If not, contact your system administrator.</li></ul> <p>Verify that your server's certificate appears on the Apple list of <a href="#">CA trusted certificates</a>.</p> <p>If the certificate is not in the list, contact your administrator to request they update the server with one of Apple's <a href="#">CA trusted certificates</a>.</p>

## Download

Download the App from Google Play (Android) and the App Store (iOS).



*Cortex XSOAR Enterprise mobile app has OS specific and app specific requirements. Ensure you meet all system requirements before downloading and installing the App.*

After downloading the App, you can [use the app](#), which has some of the same features as the web app.

## Android Certificate Requirements

You need to use a CA-signed SSL certificate from a trusted official Certificate Authority. If you cannot obtain approval to use a trusted CA, you can customize the android apk, as described in [Deploy the Android apk in a Self-Signed Certificate and an MDM Environment](#).

## Administrators

---

Before you add the CA-signed certificate to Cortex XSOAR server, ensure the certificate contains the full certificate chain. If the certificate does not contain the full certificate chain, you need to [Obtain the Full Certificate Chain for a Certificate](#).

### Users

If you do not use a trusted official CA signed SSL certificate or you do not use the [Android APK](#), you need to [Configure the Mobile Device for Users](#).

## Deploy the Android apk in a Self-Signed Certificate and an MDM Environment

You may decide to use your own private CA generated certificates for Cortex XSOAR, as opposed to buying a trusted CA certificate. If so, the Cortex XSOAR mobile app running on devices with the latest Android versions might experience difficulty connecting to the Cortex XSOAR server due to Android restrictions.



*Check whether you can connect to Cortex XSOAR through your browser, even if you cannot connect through the Cortex XSOAR app. If you cannot connect to the server through your browser, there could be other issues, such as VPN connectivity into the organization's private network.*

This procedure enables you to deploy the android apk file in an environment with a self-signed certificate and a MDM, or other internal distribution mechanism. You do this by manually changing the android apk file and allowing distribution of the apk to your users through direct link to the apk or MDM of your choice.

**STEP 1** | On a Java installed Linux or Mac computer, download the following:

1. The latest **Cortex XSOAR apk** from the Play Store or a non-signed version from the download server using your Cortex XSOAR installer download link and append `&downloadName=android_unsigned_apk` to the link.
2. The **change\_apk\_cert.sh** shell script tool from the download server using your Cortex XSOAR installer download link, and append `&downloadName=change_apk_cert` to the link.

**STEP 2** | Place the privately issued certificate (.crt file) that you wish to deploy in the Android app, on the same computer, as referred to in step 1.

**STEP 3** | Install the [APKtool](#) on the computer.

**STEP 4** | Run the script by typing the following command:

```
./change_apk_cert.sh
```

**STEP 5** | When prompted, use the other files as input.

**STEP 6** | Distribute the apk to your users (by direct link to the apk or MDM of your choice) and ensure connectivity is made.

**STEP 7** | (Optional) If the MDM environment issues an error (for example, **APK is not zip aligned, APK signature is invalid or does not exist**, or similar) you need to re-run the script with zipalign and jarsigner enabled.

1. Ensure that you install [zipalign](#), which is part of Android Studio.
2. Ensure that you install [jarsigner](#), which is part of JDK.

Ensure your machine's path is set correctly to include the jarsigner tool.

3. Run the script in step 4 and add the following options:

- 
- z, --zipalign: The path to the zipalign tool
  - k, --keystore: The path to the keystore to use for jarsigning the apk
  - a, --alias: The Alias

If the MDM environment issues an **Upload a new apk file with different package**, or a similar error, contact Customer support.

**STEP 8** | Repeat the process for every build of the apk that you wish to deploy.

## Obtain the Full Certificate Chain for a Certificate

If you are using a CA-signed certificate, ensure the certificate contains the full certificate chain. If the certificate does not contain the full certificate chain, perform these steps in your Cortex XSOAR server environment.



*Relevant for Administrators only. The following instructions are for Android 9 Pie. For other Android versions, the navigation path to find trusted credentials is different.*

**STEP 1** | Obtain an SSL certificate from a trusted Certificate Authority.

**STEP 2** | To upload the CA-signed certificate to a Cortex XSOAR server, follow the instructions in [HTTPS with a Signed Certificate](#).

**STEP 3** | (Optional) Open the certificate in `/usr/local/demisto/cert.pem`.

**STEP 4** | (Optional) Locate the root certificate and verify that it includes the `---BEGIN CERTIFICATE---` header and the `--- END CERTIFICATE---` footer.

**STEP 5** | (Optional) From the command run the following command:

```
openssl s_client -connect www.microsoft.com:443 -showcerts.
```

**STEP 6** | Copy the entire certificate chain in order.

**STEP 7** | (Optional) Paste the entire certificate chain directly under the root certificate in the cert.pem file.

**STEP 8** | (Optional) Restart the Cortex XSOAR Service

## Configure the Mobile Device for Users

If users do not use a trusted official CA signed SSL certificate or use the [Android APK](#), you need to convert the pem file to a cert file and add it to the root of your device.

**STEP 1** | In Cortex XSOAR go to **Settings > About > Troubleshooting > Security** and download the certificate.

**STEP 2** | From the command line, execute the following command:

```
openssl x509 -inform PEM -outform DM -in ~/<file_path>/<certificatename>.pem
-out ~/<file_path>/<certificatename>.cert
```

**STEP 3** | Copy the .cert file to the `/sdcard` folder on your mobile device.

---

**STEP 4** | On your mobile device, go to **Settings > Security & location > Advanced > Encryption & credentials** and when prompted, enter the certificate name for the `.cert` file.

## Use the Cortex XSOAR Enterprise Mobile App

When you **log in** to the Cortex XSOAR Enterprise mobile app you can do the following:

- [Switch Accounts in Multi-Tenants Deployments](#)
- [Manage Dashboards in the Cortex XSOAR Enterprise Mobile App](#)
- [Work with Incidents](#)
- View your own tasks in the **My Tasks** tab: chose options to complete the task and manually mark the task as complete.

### *Log in to the Cortex XSOAR Enterprise App*

If your organization requires you to connect to Cortex XSOAR using a VPN connection, ensure that you are logged into the VPN.

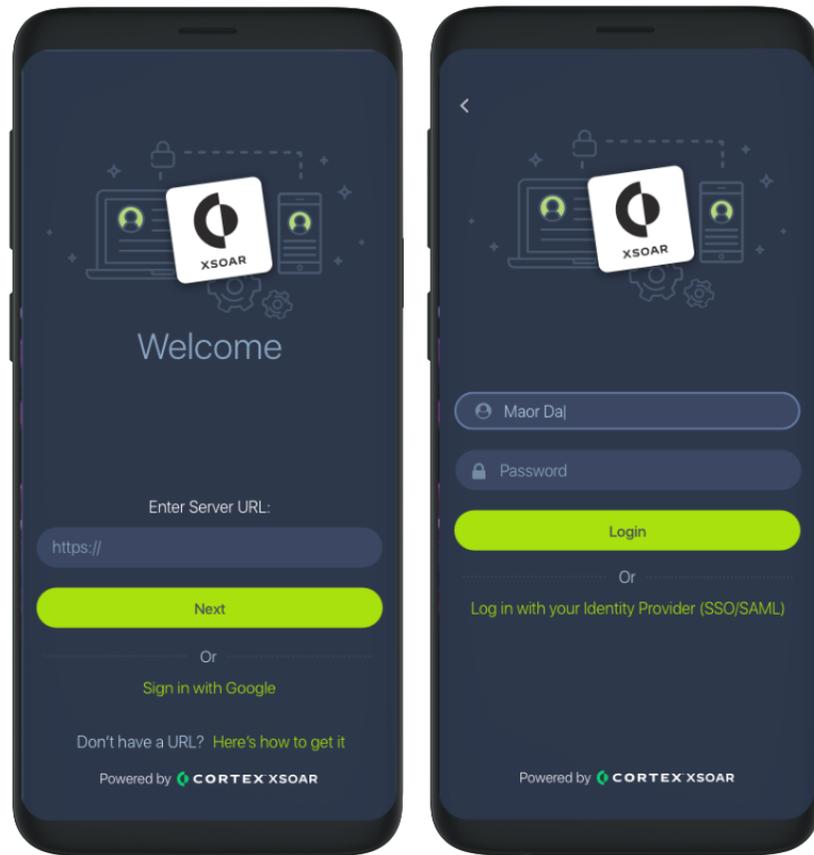
**STEP 1** | Open the Cortex XSOAR Enterprise App.

**STEP 2** | Type the Cortex XSOAR server URL.

To obtain your Cortex XSOAR server URL, open Cortex XSOAR and go to **Settings > About > Troubleshooting > Server Configuration > External Host Name** and copy the Cortex XSOAR URL.

**STEP 3** | Do one of the following:

- To authenticate with Google IAP, tap **Sign in Google**.  
Enter the username and password and tap **Next**.
- To enter your username and password or use SSO (authentication key is required) tap **Next** and enter your details.

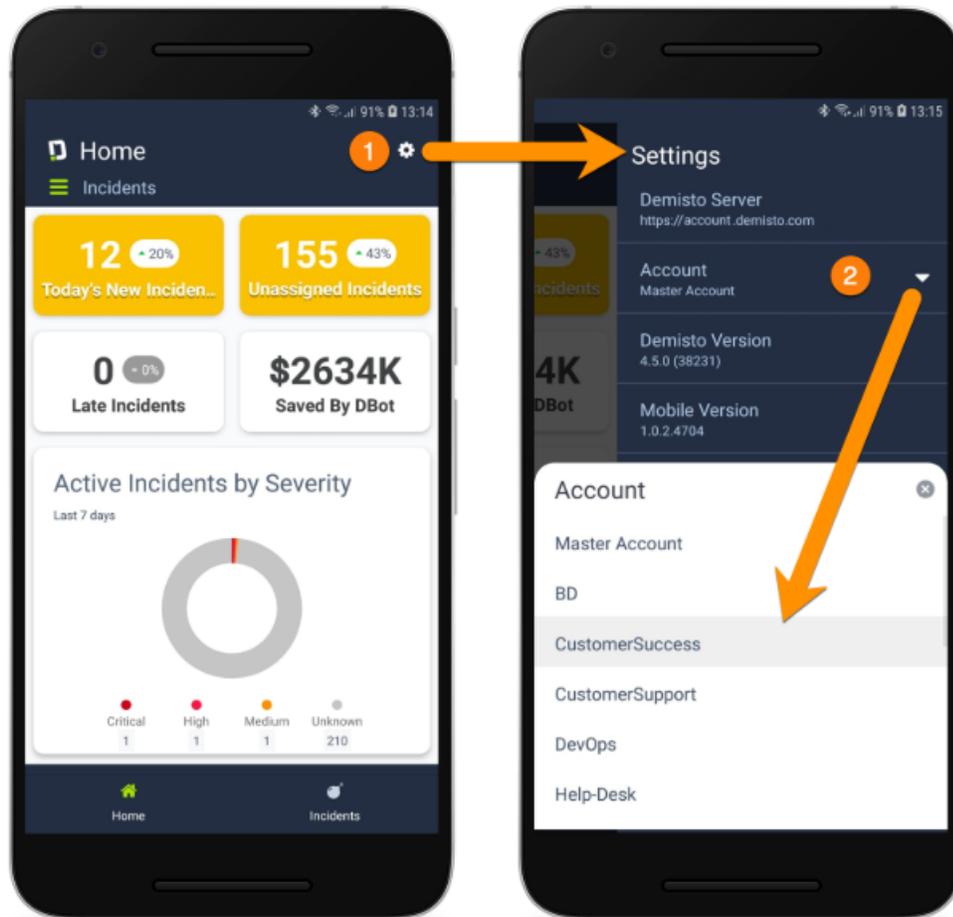


## Switch Accounts in Multi-Tenants Deployments

Similar to Cortex XSOAR, when you log in to a Multi-Tenant deployment, the Cortex XSOAR Enterprise mobile app displays the main account, where you can see dashboards and incidents for all the accounts you manage.

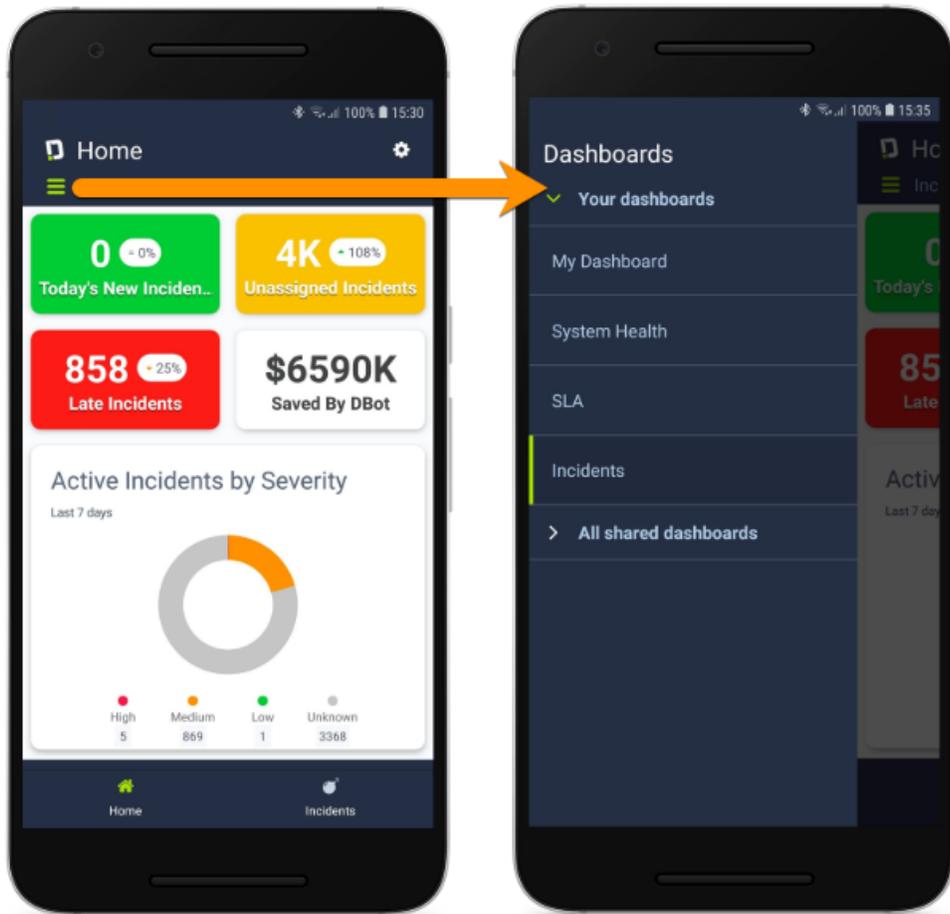
**STEP 1 |** To switch between accounts, go to **Settings > Account**.

**STEP 2 |** In the **Account** field, from the drop down list, select the account to manage.



## Manage Dashboards in the Cortex XSOAR Enterprise Mobile App

In the **Home** tab, you view your dashboard. To view all dashboards, tap  and select the dashboard you want. Like the web application, the mobile app remembers and displays the most recently viewed dashboard.



Some widgets in the dashboard open the incident list, which is automatically filtered according to the widget's filter. For example, tapping on **Today's New Incidents** widget, takes you to the **Incidents** list, showing only incidents opened today.

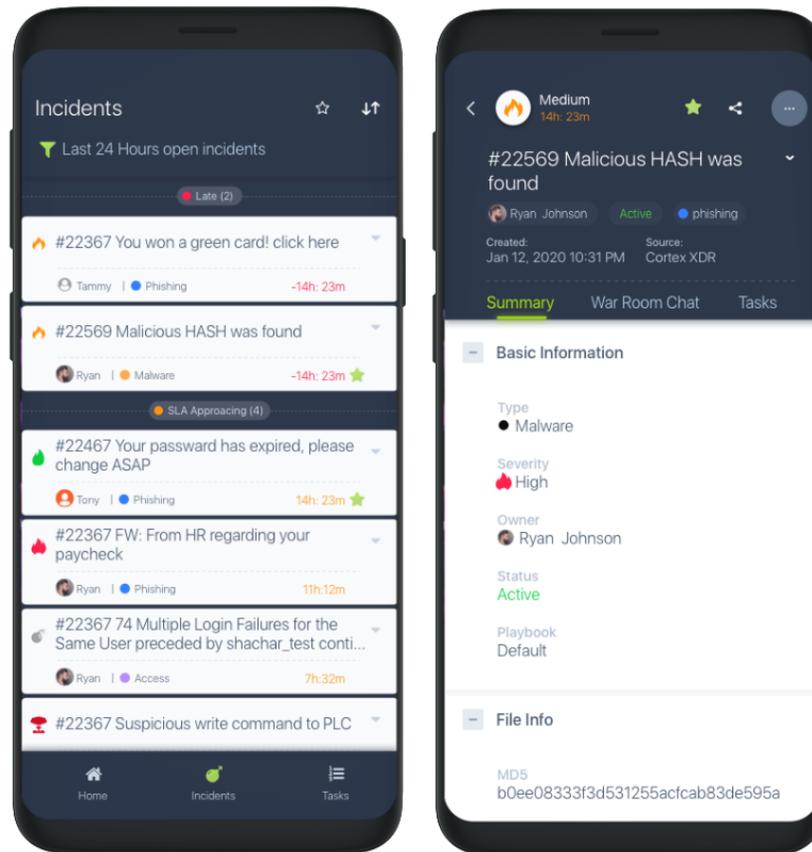
Some dashboard widgets are not supported in the mobile app. To view full dashboards, go to the web application. Use the Dashboard Builder (in the web app) to [create a custom dashboard](#) for the mobile app.

## Work with Incidents

You can view all the incidents according to your selected filter. For each incident you can view the following:

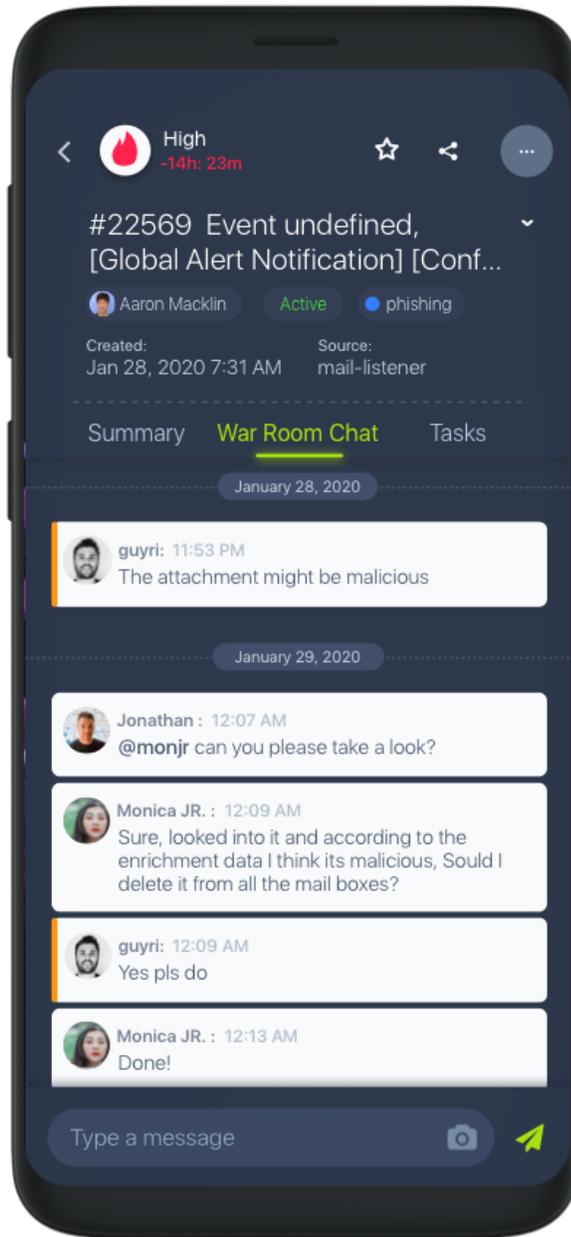
- **Summary**

The incident summary supports a limited number of sections and fields. You may want to create a custom incident summary layout for the mobile app. For more information, see [customize incident layouts](#) in the Help.



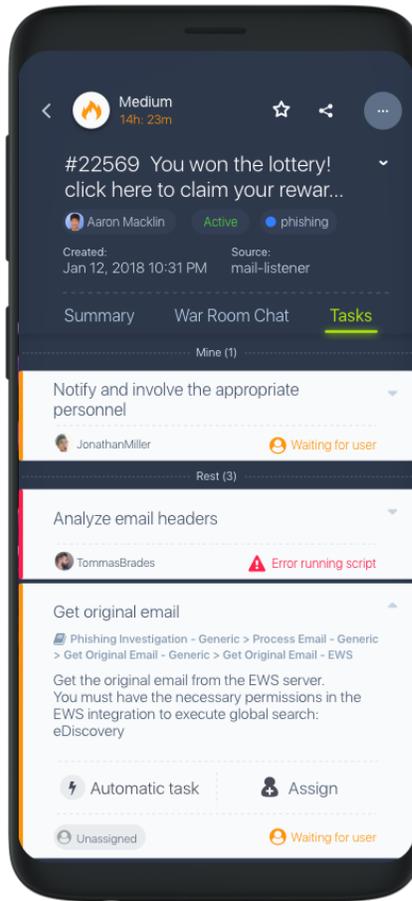
- **War Room Chat**

You can view, send and receive messages, upload, download files, and attachments.



- **Tasks**

You can assign analysts to the task, chose options to complete the task and manually mark the task as complete.



To perform action on the incident, tap . Support. Supported incident actions include:

- Assign an analyst to the incident
- Change the incident severity
- Change the incident type
- Close the incident

# Agents

- > Agents Overview
- > Shared Agents
- > D2 Agent
- > Agent Tools



---

# Agents Overview

Agents enable you to transfer files and execute commands on remote machines.

You can create a [D2 Agent](#) for a specific incident or you can create a [shared agent](#) for a number of incidents.

You need to install the **D2** Content Pack, which enables you use commands and automations for D2 and Shared Agents. You can use the automation scripts that come out-of-the box or configure and create scripts in [Agent Tools](#).



*Most D2 automations and commands are relevant for both D2 Agents and Shared Agents.*

After installation, you can run Powershell commands directly from Cortex XSOAR on common applications, such as Office 365 and Active Directory. You need to [Configure Cortex XSOAR to Use PowerShell](#), before running PowerShell commands.

---

# Shared Agents

A shared agent enables you to transfer files and execute commands on remote machines for a number of incidents. If you want to create an agent for a specific incident only, create a [D2 Agent](#).

Before installing a shared agent, you need to [create a shared agent instance](#). The default hostname must be the same as the endpoint's system name.

Shared agents enable you to do the following:

- Perform tasks from the Cortex XSOAR CLI as if you were using the target machine.
- Run pre-defined D2 Agent automation scripts.
- Create and configure automation scripts, using [Agent Tools](#).
- Run existing D2 Agent forensic tools (agent tools) as part of a Cortex XSOAR playbook.

You can run all of the D2 automations, such as D2 Exec, D2Drop, etc. You need to add the name of the shared instance at the end of each automation.

## Configure a Shared Agent Instance

To create a a shared agent, you need to configure a new integration instance for the shared agent.

**STEP 1** | Go to **Settings > Integrations > Servers & Services**.

**STEP 2** | Search for shared agent.

**STEP 3** | Create and configure a new integration instance by clicking **Add instance**.

**STEP 4** | Add [Shared Agent Instance Parameters](#).

The screenshot shows a configuration window titled "Shared Agent" with a close button (X) and a help button (?). The form contains the following fields:

- Name**: A text input field containing "sharedagent\_demo".
- Credentials**: A text input field containing "administrator". A "Switch to credentials" link is visible to the right of this field.
- Password**: A password input field with masked characters (dots).
- Default Hostname or IP Address**: A text input field containing "ec2-108-128-180-161.eu-west-1.compute.amazon".
- Target Architecture**: A text input field containing "amd64".
- Target Operation System**: A text input field containing "windows".

At the bottom of the form, there are four buttons: "Delete" (with a trash icon), "Test" (with a play icon), "Cancel", and "Done" (highlighted in orange).

**STEP 5** | Click **Done**.

You can now [Install a Shared Agent](#).

---

## Shared Agent Instance Parameters

The following table describes the shared agent parameters when creating a new instance.

Parameter	Description
Name	A meaningful name to remember for later use.
Credentials	Username for the target machine.
Password	Password for the target machine.
Default Hostname or IP Address	IP address of the target machine.
Target Architecture	amd64 or i386
Target Operating System	Windows, Linux, or OSX.
Ciphers	(Linux only) You can change the cipher mechanism used by SSH to install the agent.
Target Domain or Workgroup	Hostname (domain or workgroup) of the target machine.
SMB protocol version	For Windows remote installation SMB protocol is used (port 445). In case of SMB errors, you may need to specify SMB 2 or SMB 3.

## Install a Shared Agent

Install a shared agent on machines that are under investigation to unobtrusively perform forensic tasks on those machines.

### Before you begin:

- (Windows) You have at least Power User credentials.
- (Windows) Enable the Service Message Block Protocol.
- (Remote installations) Firewall Port 445 (SMB) is open.

If you experience issues during installation, see [Troubleshoot a Remote Installation \(Windows\)](#).

### STEP 1 | Configure a Shared Agent Instance

Verify that you have defined the external IP address or base URL of your Cortex XSOAR server by going to **Settings > About > Troubleshooting**.

### STEP 2 | If installing manually, install the shared agent on the system.

1. Type the following command:

```
!sharedagent_create system=<agent-instance_name>
```

For example, `!sharedagent_create system="sharedagent_demo"`.

2. In the Dbot response, click **Download Agent**.
3. On the target machine, unzip and run the agent zip file.
4. (Optional) In the Cortex XSOAR CLI, run the following command to test the agent installation.

---

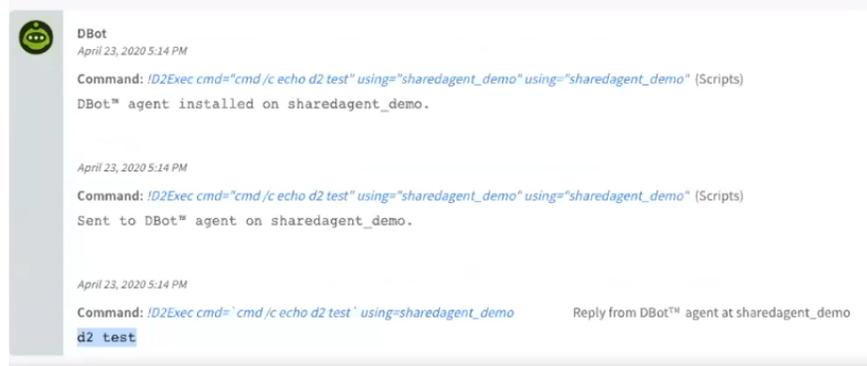
```
!D2Exec cmd=`cmd /c dir` using=agentInstanceName
```

### STEP 3 | Install the Shared Agent remotely.

The agent is installed remotely (from the Cortex XSOAR server) the first time you communicate with it.

1. Go to the incident you want to add the shared agent.
2. In the CLI, run any D2 command. For example, to test the agent installation, type the following command:

```
!D2Exec cmd="cmd /c echo d2 test" using="sharedagent-demo"
```



### STEP 4 | (Optional) Configure Agent Tools that invoke existing forensic applications.

---

# D2 Agent

Create and install Cortex XSOAR dissolvable agents (D2 agents) on machines that are under investigation to unobtrusively perform forensic tasks on those machines. After the agents complete the forensic tasks, they dissolve leaving no trace. D2 agents are designed to assist you when performing an investigation in the War Room and for a specific incident only.



*If you want to create agents for more than one incident, create a [shared agent](#)*

D2 Agents enable you to do the following:

- Create and [Install a D2 Agent](#), using the CLI. You can install remotely or manually.
- Perform tasks from the Cortex XSOAR CLI as if you were using the target machine.
- Run pre-defined D2 agent automation scripts.
- Create and configure automation scripts using [Agent Tools](#).
- Run existing D2 agent forensic tools (agent tools) as part of a Cortex XSOAR playbook.
- Kill or assign an expiration date of an agent to dissolve it on the target machine.



*D2 Agents are usually installed on Windows, as UNIX systems have different solutions, such as SSH. If you cannot access a target machine, you might need to set up a Cortex XSOAR engine before you can install and run agents on that machine.*

## Install a D2 Agent

Install a D2 agent to assist you when performing an investigation in the War Room.

Before you begin, do the following:

- **(Windows)** You have at least Power User credentials on the target machine.
- **(Windows)** Enable the Service Message Block Protocol on the target machine.
- **(Remote installations)** Firewall Port 445 (SMB) is open on the target machine.

You can install the D2 agent manually or remotely. When port 445 is open, you can install the D2 agent remotely (from the Cortex XSOAR server) the first time you communicate with it. If you experience issues during installation on Windows machines, see [Troubleshoot a Remote Installation \(Windows\)](#).

### STEP 1 | Add the system (machine under investigation) to an incident.

1. Type the following command:

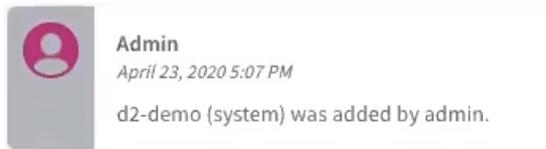
```
/system_add host=<name of the host name> arch=<name of the architecture>
os=<operating system> user=<name of user> password=<Will-Prompt-After-
Enter> name=<name of the D2 agent>
```

For example: `/system_add`

```
host=ec2-108-128-180-161.eu.west-1.compute.amazonaws.com arch=amd64
os=windows user=administrator password=<Will-Prompt-After-Enter>name=d2-
demo
```

2. Press enter, and when prompted, type the password.

In the War Room, confirmation appears that the system was added to the incident:



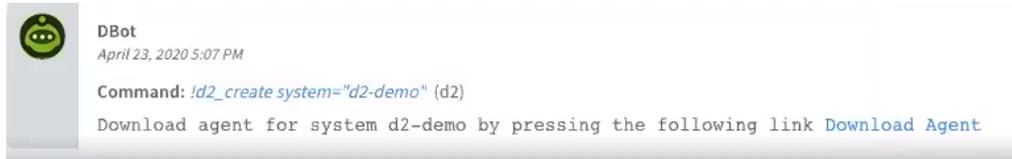
**STEP 2 |** If installing manually, install the D2 agent on the system.

1. Type the following command:

```
!d2_create system=<system_name>
```

For example, **!d2\_create system="d2-demo"**.

2. In the Dbot response, click **Download Agent**.



3. On the target machine, unzip and run the agent zip file.
4. (Optional) type the following command to test the agent installation:

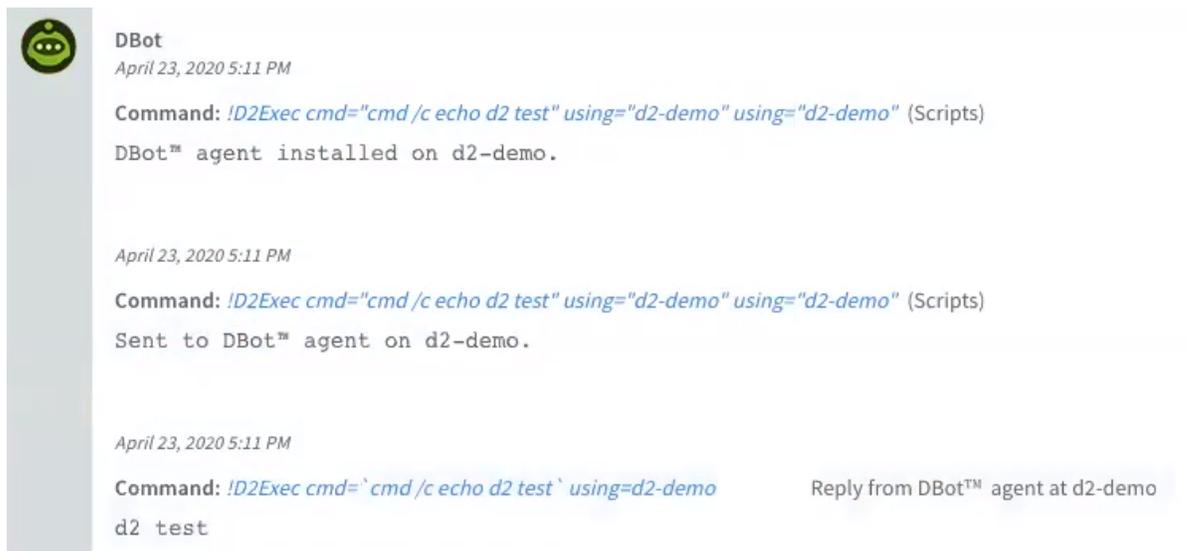
```
!D2Exec cmd=`cmd /c dir` using=<agent-instance-name>
```

**STEP 3 |** Install the D2 Agent remotely.

The agent is installed remotely (from the Cortex XSOAR server) the first time you communicate with it.

1. Go the incident you added the system in step 1.
2. In the CLI, run any D2 command. For example, to test the agent installation, type the following command:

```
!D2Exec cmd="cmd /c echo d2 test" using="d2-demo"
```



**STEP 4 |** (Optional) Configure **Agent Tools** that invoke existing forensic applications.

---

# Troubleshoot a Remote Installation (Windows)

The following table describes the error messages you may receive when remotely installing the shared agent or D2 agent on Windows.

Error Message	Workaround
<code>NT_STATUS_LOGON_FAILURE</code>	Verify that the username and password are correct in the integration instance configuration settings.
<code>NT_STATUS_NO_MEMORY</code>	<ul style="list-style-type: none"><li>• This error is not related to memory. The workgroup is missing.</li><li>• If the target machine is inside the domain, enter the domain in the <b>Target domain or workgroup</b> parameter in the integration instance configuration settings.</li><li>• If the target machine is not in the domain, enter the machine's hostname (not the host address) in the <b>Target domain or workgroup</b> parameter in the integration instance configuration settings. To get the machine's hostname, run the hostname command from the terminal.</li></ul>
<code>NT_STATUS_UNSUCCESSFUL</code>	The IP address is incorrect.
<code>NT_STATUS_IO_TIMEOUT</code>	<ul style="list-style-type: none"><li>• Firewall is blocking SMB/Port 445.</li><li>• Wrong address</li></ul>
<code>NT_STATUS_DUPLICATE</code>	This error is related to a DNS issue. If you are using Amazon, use the actual IP address and not the URL.
<code>NT_STATUS_CONNECTION_RESET</code>	The target machine might not support SMB 1 connection. Make sure SMB2 is active on the target machine and specify the SMB argument value as 2. For more information about how to detect, enable, and disable different versions of SMB for Windows and Windows server, see <a href="#">SMB assistance</a> .
Agent is installed but unresponsive	Verify that the base URL for D2 agents and engines is correct and reachable from the network segment where the agent is installed. Go to <b>Settings &gt; About &gt; Troubleshooting</b> and verify that you defined the external IP address or base URL of your Cortex XSOAR server.

---

# Agent Tools

Agents make use of pre-defined end user generated automation scripts. You can extend scripting functionality by deploying forensic agent tools with a Cortex XSOAR Agent.

Agent Tools comes with a number of out of the box scripts, which can be configured, such as Office365, Active Directory, and WinPmem.

You can create your own scripts and files by going to **Settings > Agent Tools > + Add Tool**. The files and scripts must be in zip, tar.gz, tar.bz2, or tar format.

For example, you can create the following scripts:

- [Run a batch file.](#)
- [Return the memory dump to the War Room.](#)
- [View all processes](#)

Once deployed, the agent can use the tool (e.g. create a memory dump can be copied to another machine for forensic analysis).

Although you can run PowerShell commands directly from Cortex XSOAR on applications such as Office 365 and Active Directory, if you want to use PowerShell scripts, you need to [Configure Cortex XSOAR to Use PowerShell](#) Use the [D2 Agent Script commands](#) to assist you with script arguments.

## Configure Cortex XSOAR to Use PowerShell

You can run PowerShell commands directly from Cortex XSOAR on common applications such as Office 365 and Active Directory. If you want to use PowerShell, you need to configure Cortex XSOAR.

Relevant for both D2 agents and shared agents.

### STEP 1 | Create the PowerShell script you want to run.

In this example, we have created a PowerShell script, called *printarg* to print an argument.

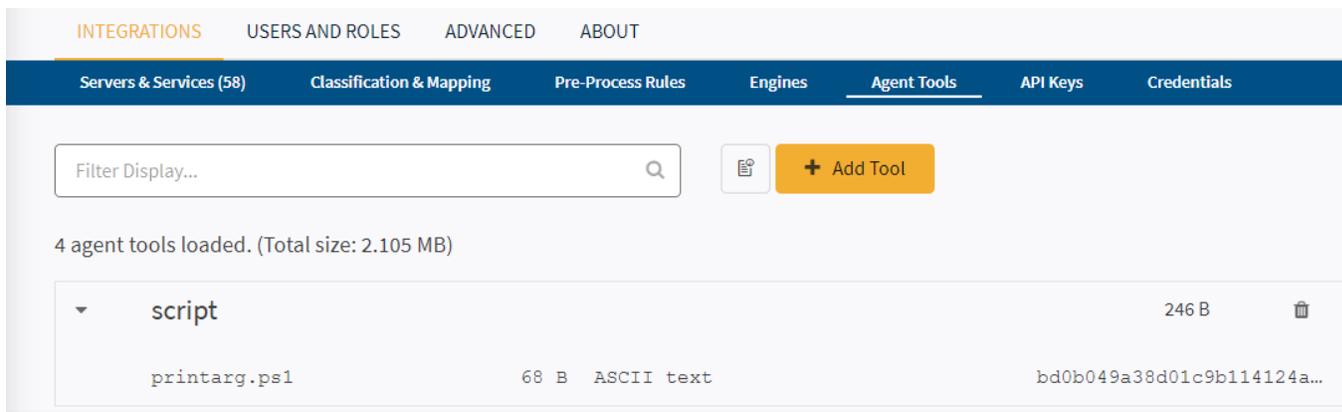
```
param([string]$myarg = "")
Write-Host "This is my argument: " $myarg
```

1. Zip up the file. In this example, we will call the file *script*.

### STEP 2 | Upload the zip file you created in step 1.

1. Select **Settings > Integrations > Agent Tools > Add Tool**
2. Add the file.

You can see the script file contains the PowerShell script.



### STEP 3 | Create an automation that runs the PowerShell script.

1. Go to the **Automation** page and upload the script.

In this example we have created an automation called D2PowerShellEx.

```
script: |
 //+ script/printarg.ps1

 if ((env.ARCH !== "amd64") && (env.OS !== "windows")) {
 throw("Script can run only in 64bit Windows Agents");
 }
 var command = [];
 command.push("powershell.exe");
 command.push("-NonInteractive");
 command.push("-NoLogo");
 command.push("'" + which("printarg.ps1") + "'");
 if (typeof (args.myarg) !== "undefined") {
 command.push("-myarg");
 command.push(args.myarg);
 }
 timeout = 60 * 5;
 if (typeof (args.timeout) !== "undefined") {
 timeout = args.timeout;
 }
 var results = execute(command.join(" "), timeout);//, 'table');
 pack('Stdout: ' + results.Stdout + '\nStderr: ' + results.Stderr);
type: javascript
tags: []
enabled: true
args:
- name: myarg
 required: true
- name: timeout
scripttarget: 1
timeout: 7.2µs
pswd: ""
runonce: false
runas: DBotWeakRole
```

Important to note:

**command.push("powershell.exe"):** Runs the PowerShell.

**command.push("'" + which("printarg.ps1") + "'"):** The absolute path of the executable script.

**//+ script/printarg.ps1:** Annotation that tells the agent which tools to send to the Windows machine. The name of the zip file (**script**) and the script name (**printarg.ps1**).

- For more information about the commands, see [D2 Agent Script Commands](#).
2. Click Save.

#### STEP 4 | Run the automation in the War Room.

To run the automation, you need to install either a [D2 Agent](#) or a [Shared Agent](#)

For example, type `!D2PowerShellEx myarg='success' using=sharedagent-demo`

```

DBot
April 23, 2020 5:21 PM
Command: !D2PowerShellEx myarg='success' using=sharedagent_demo using=sharedagent_demo (Scripts)
Sent to DBot™ agent on sharedagent_demo.

April 23, 2020 5:21 PM
Command: !D2PowerShellEx myarg='success' using=sharedagent_demo
Reply from DBot™ agent at sharedagent_demo
Stdout: This is my argument: success
Stderr:

```

## D2 Agent Script Commands

The following are the D2 agent scripting commands. Each command is followed by its description, its syntax, and an example where applicable.

 *Cortex XSOAR server comes with a few example agent scripts. These will help you get more acquainted with the functions. You can copy the scripts, change them and check the results.*

Command	Syntax	Description
<b>env</b>	<code>var env = {OS:"",ARCH:""};</code>	Holds the environment variables. env.OS and env.ARCH are populated with OS and architecture.  Example: <code>console.log(JSON.stringify(env))</code>
<b>pwd</b>	<code>function string pwd();</code>	Returns the absolute path of the working folder.
<b>which</b>	<code>function string which(path string);</code>	Returns the absolute path for a given path or executable.  Example: <code>console.log(which('ls'));/bin/ lsconsole.log(which('syslog'));/ usr/bin/syslog</code>
<b>execute</b>	<code>function ExecResult execute(cmd string);</code>	Executes the given command.  Returns:  {Stdout string // process stdout captured Stderr string // process stderr captured PID int // PID of process that was running Success bool // whether process ended successfully Error string // string describing the error if exists}

Command	Syntax	Description
		Example: <code>var ret=execute('ls -l'); console.log(ret.Stdout); console.log(JSON.s</code>
<b>pack</b>	function null pack(content object, contentformat string[optional]);	Returns the content as an entry on the investigation. Content can be a JSON object or when specified value. <i>contentformat</i> may be one of the following: 'table', 'text' or 'json'. If not provided, the format will be determined according the type of content.
<b>pack_file</b>	function null pack_file(path string, content string[optional]);	Returns the <i>path</i> as a file entry on the investigation. If content is provided, it will be attached to the file.
<b>files</b>	function []FileInfo files(folder string, recurse bool[=false], hashes bool[=false], regex string[=""]);	Retrieves a list of files from the folder. If recurse is true, sub-folders will be included. If hashes is true, it will compute hashes for each file. If regex is provided, it will return only file names matching the regex.  Returns an array of: {Created int CreatedStr string Accessed int AccessedStr string Changed int ChangedStr string Path string Type string Size int Mode string MD5 string SHA1 string SHA256 string SHA512 string SSDeep string}  Example: <code>console.log(JSON.stringify(files('/tmp', true, true)));</code>
<b>copy</b>	function int copy(src string, dest string, overwrite bool[=false], regex string[=""]);	Copies the source (src) to the destination (dest). If overwrite is false, it will throw an exception if the destination exists. If regex is provided, it will copy only files matching the regex. This function is not recursive.  Returns: The number of items copied.
<b>move</b>	function int move(src string, dest string, overwrite bool[=false], regex string[=""]);	Same as copy, but also deletes the source files.
<b>del</b>	function int del(file string, regex[=""]);	Deletes the file. If the file is a folder, and regex is not empty, it will remove only the files matching regex from that folder.
<b>grep</b>	function []GrepMatch grep(path string, regex string, recursive bool[=false]);	Searches the given path for files matching regex. If recursive is true, it will dive into the sub folders.  Returns an array of: { Path string // Path to file matching Offsets [][]int // The matching indexes on the line}

Command	Syntax	Description
		<p>Example:</p> <pre>console.log(JSON.stringify(grep( '/tmp/', 'Scan', true )));</pre>
<b>strings</b>	function []string strings(path string, min int[=4], max int[=1024]);	<p>Searches strings contained in the file provided by path. Use min and max to control the sizes of the strings that are captured.</p> <p>Example:</p> <pre>console.log(JSON.stringify(strings( '/bin/ls' )));</pre>
<b>bytes</b>	function string bytes(file string, offset int[=0], size int[=1024]);	<p>Returns a <i>size</i> bytes chunk of a <i>file</i> starting at <i>offset</i>.</p> <p>Example:</p> <pre>console.log(JSON.stringify(bytes( 'ddb', 0, 15 )));</pre>
<b>mkdir</b>	function bool mkdir(path string);	<p>Returns 'true' if a folder was created. Throws an exception otherwise.</p>
<b>rmdir</b>	function bool rmdir(path string);	<p>Removes the folder provided by *path.</p> <p>Returns: 'true' if a folder was removed. Throws an exception otherwise.</p>
<b>join_path</b>	function string join_path(part1, part2... string);	<p>Joins the paths provided by <i>part1</i> to <i>partN</i>.</p> <p>Returns: Path string.</p> <p>Example: <code>console.log(join_path( "/tmp", "one", "two", "three.file" )); /tmp/one/two/three.file</code></p>
<b>http</b>	function HTTPResponse http(url string, arg object);	<p>Performs HTTP GET call to URL with the provided arg as a request body.</p> <p>Returns object: {StatusCode int // HTTP response code Status string // HTTP status as text Cookies []http.Cookie Body string Headers string[][]}</p> <p>http.cookie object: Name string Value string Path string // optional Domain string // optional Expires time.Time // optional RawExpires string // for reading cookies only // MaxAge=0 means no 'Max-Age' attribute specified. // MaxAge&lt;0 means delete cookie now, equivalently 'Max-Age: 0' // MaxAge&gt;0 means Max-Age attribute present and given in seconds MaxAge int Secure bool HttpOnly bool Raw string Unparsed []string // Raw text of unparsed attribute-value pairs</p> <p>Example:</p> <pre>console.log(JSON.stringify(http( "http://www.google.com/lala" )));</pre>

Command	Syntax	Description
<code>read_file</code>	<code>function string read_file(path string);</code>	Returns the entire content of the path. Throws an exception if it does not exist.
<code>wait</code>	<code>function string wait(seconds int);</code>	Sleeps for the number of defined seconds.

### Windows Specific Functions

Command	Syntax	Description
<code>processes</code>	<code>function ProcessInfo[] processes();</code>	Returns a list of processes.
<code>services</code>	<code>function ServiceInfo[] services();</code>	Returns a list of services.
<code>wmi_query</code>	<code>function Object[] wmi_query(query string);</code>	Executes a WMI query. Returns an array in JSON representing the results.
<code>registry</code>	<code>function Object[] registry(path string);</code>	Gets all values under the registry path provided by <i>path</i> as a set of JSON objects. This function is always recursive if a key name is provided.  The key name must start with one of the following: "HKEY_CLASSES_ROOT", "HKEY_CURRENT_USER", "HKEY_LOCAL_MACHINE", "HKEY_USERS" or "HKEY_CURRENT_CONFIG".
<code>ifconfig</code>	<code>function Object[] ifconfig();</code>	Returns a list of all interface adapters and their configurations.
<code>fsconfig</code>	<code>function Object[] fsconfig();</code>	Returns a list of all file systems.
<code>accounts</code>	<code>function Object[] accounts();</code>	Returns a list of all defined user accounts.

## Return the memory dump file script

You can run files using the D2Winpmem automation, which returns the memory dump file to the War Room. Useful when dealing with any type of malware. You can use this for both shared agents and D2 agents.

**STEP 1** | Go to the **Automations** page and search for **D2Winpmem** automation.

**STEP 2** | Select **Copy Script**.

**STEP 3** | In the `//+winpmem/winpmem_2.0.1.exe` line in the script, change it to the file you want to run. For example, `//+New-collectorD2/New-collectorD2.bat`

**STEP 4** | In the `var exename = 'winpmem_2.0.1.exe';` line write the file you want to execute.

---

**STEP 5** | In the `var dumpFile` add the file you want to run.

```
//+New-collectorD2/New-collectorD2.bat
// {
 if (env.OS !== 'windows') {
 throw ('script can only run on Windows');
 }
 var arch = wmi_query('select OSArchitecture from win32_operatingsystem')
[0].OSArchitecture;
 var exename = 'Testd2.bat';
 var dumpFile = env.TEMP+ '\\New-collectorD2.bat';
 var output = execute('cmd /c dir /s ' + env.TEMP , 30); // 10 minutes
timeout
 pack(output);
 //if (output.Success) {
 // pack_file(dumpFile);
 // del(dumpFile);
 // } else {
 // throw output.Error;
 // }
 // pack('Winpmem failed: ' + ex);
 // }
```

**STEP 6** | Click **Save**.

## Running a Batch file Using Agent Tools

Run a simple batch file that returns results of a `dir` command. You can use this Automation either in a Playbook or in the Cortex XSOAR CLI (manual investigation in an incident War Room).

### Before you begin:

- Open ports between the Cortex XSOAR server and the Windows server:
  - Port 445 from Cortex XSOAR server to Windows server.
  - Port 443 from Cortex XSOAR server to Windows server and vice versa.
- Set credentials for the Windows server

**STEP 1** | Create a batch file.

The file must be in ZIP or Tar format.

In this example, we add created a batch file, called TestBatch, containing the following.

```
cd c:\
dir
```

**STEP 2** | Upload the batch file to run.

1. Select **Settings** > > **Agent Tools** > + **Add Tool**
2. Drag-and-Drop or browse to the Zip or Tar file created in step 1.

**STEP 3** | Add a system to the incident in the CLI or Automation.

Use the following automation called "D2Execute.yml" to install the D2 Agent from within a playbook and run the automation (D2Run) that is running the utility/batch.

```

commonfields:
 id: ef9edd54-0580-4945-8f06-f43dfb69fb20
 version: 20
 name: D2Execute
 script: |-
 demisto.results(demisto.executeCommand("addSystem",
{"name":demisto.args()["name"], "host":demisto.args()["host"],
 "username":demisto.args()["username"], "password":demisto.args()
["password"], "os":demisto.args()["os"]}))
 demisto.results("Installing Agent...")
 demisto.results(demisto.executeCommand("d2_install",
{"system":demisto.args()["name"]}))
 demisto.results("Running script...")
 demisto.results(demisto.executeCommand(demisto.args()["scriptName"],
{"id":demisto.args()["name"], "using":demisto.args()["name"]}))
 type: python
 tags: []
 enabled: true
 args:
 - name: name
 required: true
 default: true
 description: System name
 - name: host
 required: true
 description: Computer name
 - name: os
 required: true
 auto: PREDEFINED
 predefined:
 - linux
 - osx
 - windows
 description: OS
 - name: username
 required: true
 description: username
 - name: password
 required: true
 secret: true
 description: password
 - name: scriptName
 required: true
 description: Script Name
 scripttarget: 0

```

#### STEP 4 | Execute the utility running the CLI or Automation.

For example, use the following D2Run.yml automation:

```

commonfields:
 id: 9a18460a-e72f-488a-8112-044c9a7be76a
 version: 13
 name: D2Run
 script: |-
 //+TestBatch/TestBatch.bat

 var batch_file = 'TestBatch.bat';

```

```

if (env.OS !== 'windows') {
 throw ('script can only run on Windows');
}

var d2path = pwd();
var batch_path = d2path + '\\\' + batch_file;
batch_path = batch_path.replace(/\\\/g, "\\\\\\\\\\\\\\");

pack(execute('cmd /c ' + batch_path, 60));
type: javascript
tags: []
enabled: true
scripttarget: 1

```

Note the following:

`//+TestBatch/TestBatch.bat`: this is the name of the zip/batch file that you upload in Agent Tools.

`var batch_file = 'TestBatch.bat'`; name of the batch file to run.

## View All Running Processes Script

You can run the **D2Processes** automation to see all the running processes.

**STEP 1** | Go the incident where you want to run the D2 Processes automation.

**STEP 2** | Type the following command:

**!D2Processes using=<name of agent instance>**

For example, **!D2Processes using=sharedagent\_demo**

CommandLine	Description	Handle	ProcessId	ThreadCount	WorkingSetSize
	System Idle Process	0		8	4096
	System	4	4	126	86016
	smss.exe	340	340	2	937984
	csrss.exe	420	420	10	3768320
	csrss.exe	488	488	9	3194880
	wininit.exe	512	512	1	3633152
winlogon.exe	winlogon.exe	552	552	2	11161600
	services.exe	624	624	4	6639616
C:\Windows\system32\lsass.exe	lsass.exe	640	640	7	15548416
C:\Windows\system32\svchost.exe -k DcomLaunch	svchost.exe	724	724	23	20770816
C:\Windows\system32\svchost.exe -k RPCSS	svchost.exe	784	784	10	10743808
C:\Windows\system32\svchost.exe -k netsvcs	svchost.exe	908	908	33	65056768
C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted	svchost.exe	936	936	18	22298624
C:\Windows\System32\svchost.exe -k termsvcs	svchost.exe	988	988	51	94433280
C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted	svchost.exe	376	376	12	20533248
C:\Windows\system32\svchost.exe -k LocalService	svchost.exe	372	372	17	17473536
C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted	svchost.exe	860	860	6	5574656
C:\Windows\system32\svchost.exe -k NetworkService	svchost.exe	352	352	25	24109056
"dwm.exe"	dwm.exe	1112	1112	17	30003200
C:\Windows\system32\svchost.exe -k LocalServiceNoNetwork	svchost.exe	1160	1160	21	19640320

Partial View: Showing 20 out of 50 rows. [View full table in a new tab.](#)

# Logs

- > Audit Trail
- > Send the Audit Trail to an External Log Service



---

# Audit Trail

The audit trail displays a log of all administrative user interactions with Cortex XSOAR. The log is sorted by date and covers which users interacted in what way with system objects, and associated data. The audit trail does not include actions performed in the war room. These actions are documented in the war room.

You can search the audit trail log for user interactions based on free text.

To view an audit trail, navigate to **Settings > Users and Roles > Audit Trail**.

To customize which columns are visible in the audit trail log, click the table settings button.

To export the audit trail log, use the **GetAudits** command from the Cortex XSOAR REST API. See the Cortex XSOAR REST API documentation.

## Extract a Day's Audit Trail

You can write a script that runs daily to extract that day's audit trail, and upload it to your SIEM with uploader programs. The following is an example of a curl command that will fetch all audits from June 22, 2017 and later - up to 10,000 actions.

```
curl -k -X POST https://<IP>:<PORT>/settings/audits -H 'accept: application/json' -H 'authorization: <API KEY>' -H 'content-type: application/json' -d '{"size" : 10000,"query": "modified:>2017-06-22T00:00:00"}'
```

## Purge Audit Entries

You can define the retention period of the audit trail. By default audit entries will be retained forever. To purge periodically, add a server setting in **Settings > About > Troubleshooting** where the key is:

- **demisto.audits.purge** (true will start the purging process)
- **demisto.audits.purge.retention**. The value is the number of days to save the log. Default is 365.

To define how often to check the audit trail log, in **Settings > About > Troubleshooting** add **demisto.audits.purge.delay** where the value is how often to run the retention (**demisto.audits.purge.retention**). The default is every 24 hours.

Purging can also be done manually. The following is an example of a curl command that will purge all audits from June 22, 2017 to June 30, 2017.

```
curl -k -X POST https://<IP>:<PORT>/settings/audits/purge -H 'accept: application/json' -H 'authorization: <API KEY>' -H 'content-type: application/json' -d '{"page": 0, "size": 100,"fromDate": "2017-07-22T09:01:08.462954465+03:00","toDate": "2017-07-30T12:23:08.462954597+03:00","period": {"by": "", "toValue": null, "fromValue": null, "field": "" }, "fromDateLicense": "0001-01-01T00:00:00Z"}'
```

---

# Send the Audit Trail to an External Log Service

Organizations that are required to implement centralized logging for compliance and monitor requirements will benefit from the syslog export.

To send the Cortex XSOAR audit trail to an external log service, you create several custom server configurations.

**STEP 1** | Select **Settings > About > Troubleshooting > Add Server Configuration**.

**STEP 2** | Add the following required keys and values:

Key	Value	Required
<code>syslog.enabled</code>	<code>true</code>	Required
<code>syslog.protocol</code>	<code>tcp</code> <code>udp</code> <code>tcp+tls</code> <code>unix</code> Default: <code>unix</code> (for localhost syslogging)	Required
<code>syslog.host</code>	<code>&lt;syslog server&gt;</code>	Required. This key is not required for UNIX protocol.
<code>syslog.port</code>	<code>&lt;destination port&gt;</code>	Required. This key is not required for UNIX protocol.
<code>syslog.tag</code>	<code>&lt;syslog tag&gt;</code> Default is Demisto.	Optional
<code>syslog.format</code>	RFC3164 RFC5424 Default is empty, which uses a custom format that is a combination of both formats.	Optional
<code>syslog.filter</code>	<code>&lt;object/action&gt;</code>	Optional
<code>syslog.priority</code>	Default is a number for LOG_INFO with LOG_DAEMON. If you want to change the default, contact Demisto Support	Optional

## Filter Example

In this example, we want to match audit trail entries of `login success` and `login failure`. To accomplish this, we set the `syslog.filter` parameter to `login/.*`.

## Sample Syslog

```
CEF:0|Demisto|Demisto Enterprise|3.6.0-
master.27665.da330b76ddbdf9bbf8eldab82978550f2b5446c8|login|
failure|3|suser=john startTime=1521835930684963 cs1=john
 cs1Label=identifier cs2=Cookie: [hsfirstvisit\=http%3A%2F%2Flocalhost
%3A7070%2F||1464113050169; __ga\=GA1.1.296351272.1461620234; __hstc
\=181257784.72e717e61ded2087506747f5da786796.1464113050173.1464122836548.1469409642245.3
__hssrc\=1; hubspotutk\=72e717e61ded2087506747f5da786796; local-
theme\=light; XSRF-TOKEN\=AgF4Peo0Gqa8z1IB31fL19rHrbsLn2uri/zJ
+cWVdfE06qS+e59V3iNhJ1uhCFsPQ+N+s7nIXmvz6coWudI0/XTPHC5q4nZeyZNMl+8/
u4nQlWQEPNX2Go2Rsbny3J8b14Je4Ch8mMfVE2ES69TcLkE1Vq3/UaIQdkBV3v9IyCY
\=]\n\nReferer: [http://localhost:8080/]\n\nAccept: [application/
json]\n\nX-Xsrf-Token: [AgF4Peo0Gqa8z1IBk1GL19rHrbsLn2uri/zJ
+cWVdfE06qS+e59V3iNhJ1uhCFsPQ+N+s7nIXmvz6coWudI0/XTPHC5q4nZeyZNMl+8/
u4nQlWQEPNX2Go2Rsbny3J8b14Je4Ch8mMfVE2ES69TcLkE1Vq3/UaIQdkBV3v9IyCY
\=]\n\nContent-Length: [142]\n\nAccept-Language: [en-US,en;q\=0.9,he;q
\=0.8,ms;q\=0.7]\n\nContent-Type: [application/json]\n\nhost/ip:
[::1]:62296\nConnection: [keep-alive]\n\nUser-Agent: [Mozilla/5.0
(Macintosh; Intel Mac OS X 10_13_3) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/65.0.3325.162 Safari/537.36]\n\nApi_truncate_results: [true]\n
\nOrigin: [http://localhost:8080]\n\nAccept-Encoding: [gzip, deflate,
br]\n\nerror: Invalid credentials:Invalid username or password\nDnt: [1]\n
cs2Label=details
```

```
CEF:0|Demisto|Demisto Enterprise|3.6.0-
master.27665.da330b76ddbdf9bbf8eldab82978550f2b5446c8|login|in|3|suser=john
startTime=1521835934123052 cs1=john cs1Label=identifier cs2=host/ip:
[::1]:62296\nUser-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_3)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.162 Safari/537.36
cs2Label=details client: 127.0.0.1:57284
```

```
message: CEF:0|Demisto|Demisto Enterprise|3.6.0-
master.27665.da330b76ddbdf9bbf8eldab82978550f2b5446c8|login|out|3|
suser=john startTime=1521835927224046 cs1=john cs1Label=identifier cs2=N/A
cs2Label=details
```

