Configuration

The basic concept behind SSL decryption is to generate a CA which allows the PA to act as a SSL forward proxy between the client and the destination server.

This CA must be imported into the clients certificate store (keep in mind that Firefox does not use the system wide certificate store, it has it's own store).

The client requests an encrypted webpage and the PA generates a "fake" certificate for that site while opening another connection on behalf of the client towards the destination server. The client will e.g. call https://www.google.com. Looking into the certificate chain on the clients broiwser will reveal that this certificate was created and signed by the PA CA.

In order to make this all work, one needs to do several steps.

Selfsigned CA

This article only covers self signed certificates, not the way to enable ssl decryption with official CA certificates

Avoid white spaces

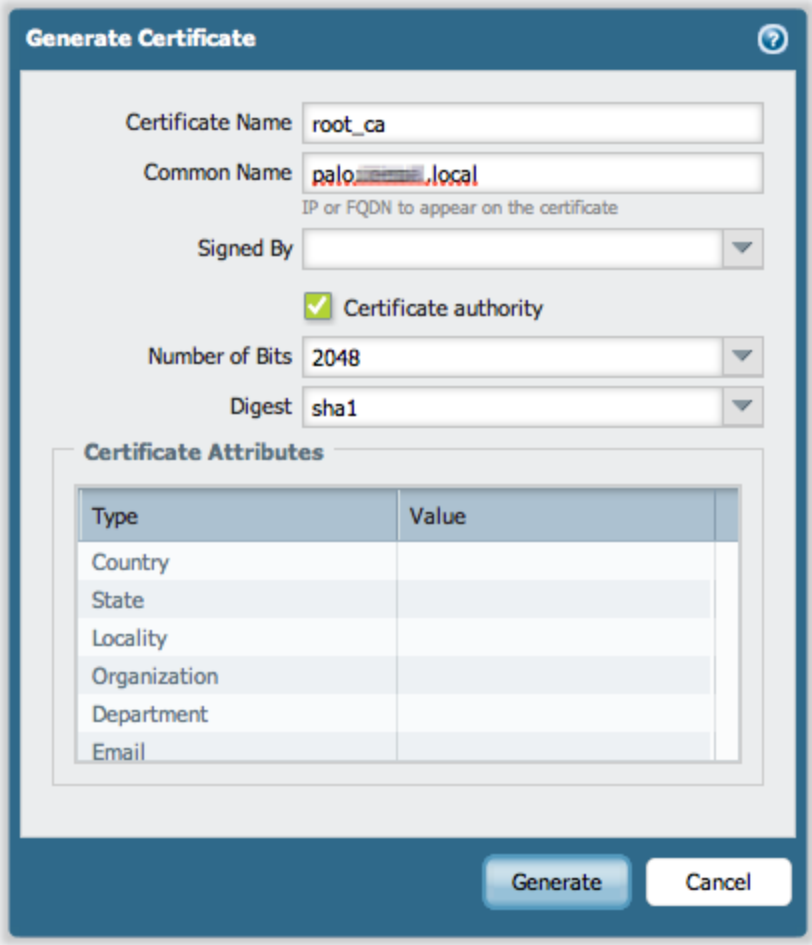
The PA behaves strange with white spaces, so you should not use any white spaces in certificate names or Common Names

Creating the certificates

Creating a root CA

It's a good idea to create a root ca which does nothing else than signing other certificates (even other CA's as we see later). Doing so allows to use certificates in several places (authentication, NTML, Global Protect, etc.) with the need of importing only this specific CA.

Go to "Device" --> "Certificates" and click "Generate"

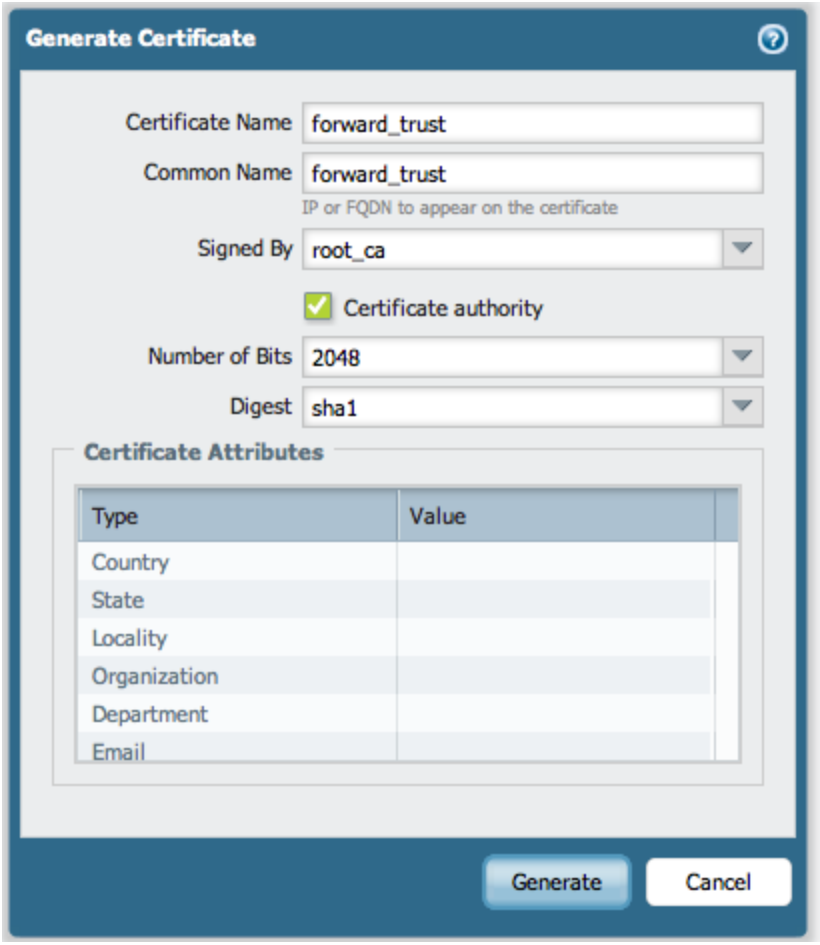


- Give it a descriptive name
- assign a common name (best to use the FQDN of the PA)
- enable "Certificate Authority"
- you can (but don't have to) fill out the Certificate Attributes

Create a Forward-Trust CA

Next we create a Forward-Trust certificate. This will be used to create the certificates for the websites the client browses if - and only if - the original certificate on the destination server is a valid certificate.

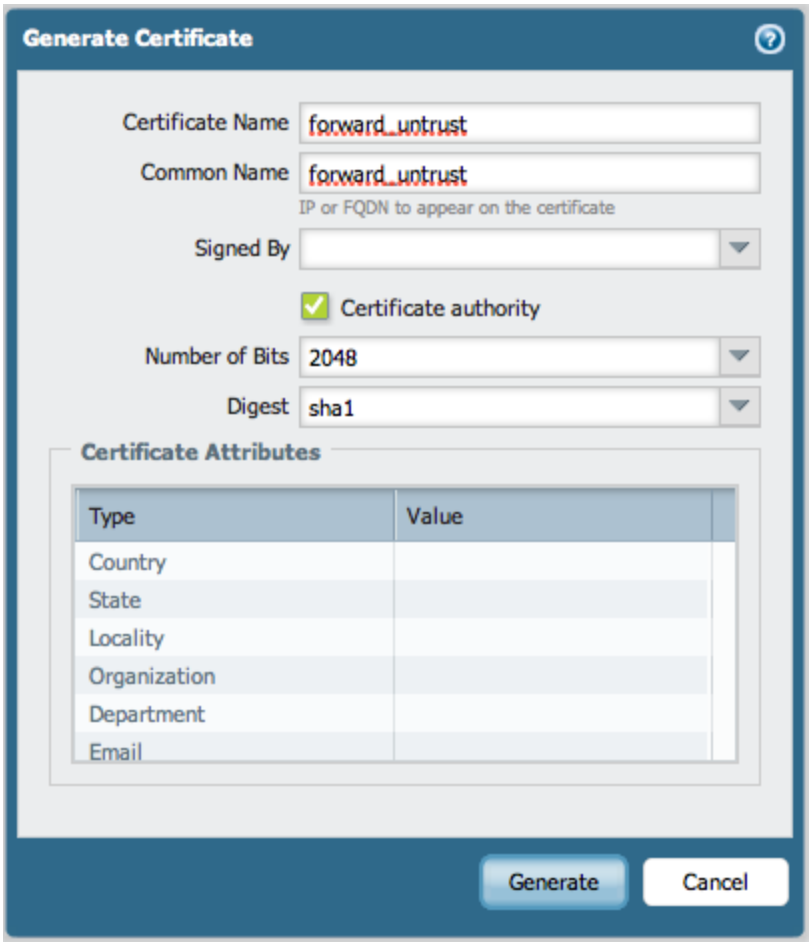
Click "Generate" again



- give a descriptive name
- give it a common name (you can use the FQDN here but I've chossen to give it a more descriptive name here as it makes troubleshooting easier. You can assign the FQDN, but the you should give it the tag "forward_trust" in the Certificate Attribute list somewhere)
- choose the root CA you've already created to sign this certificate
- make it a CA
- you can (but don't have to) fill out the Certificate Attributes

Create a forward-untrust CA

Next we create a forward-untrust CA. This will be used when the client calls a SSL URL where the original destination servers certificate is NOT valid. We'll make this as it's own CA, which will NOT be imported in the clients certificate store, allowing the browser to show SLL warnings when this certificate is used, i.e. the original certificate on the destionation server is invalid.



- give a descriptive name
- give it a common name (you can use the FQDN here but I've chossen to give it a more descriptive name here as it makes troubleshooting easier. You can assign the FQDN, but the you should give it the tag "forward_trust" in the Certificate Attribute list somewhere)
- Do NOT choose any certificate in the "Signed by" field
- make it a CA
- you can (but don't have to) fill out the Certificate Attributes

Exporting the certificates

Now you need to export the following certificates and import them into the client (See PAN Knowledgebase for information about using Windows Domain policies to automate this step)

- root_ca
- forward_trust

Do NOT import the "Forward_untrust" certificate into the clients certificate store!

Configuring certificate usage

During the creation of the certificates you cannot specify they're usage. After you created them, click on the forward-untrust certificate and mark "Forward Untrust Certificate" and make the forward-trust certificate for "Forward Trust Certificate" usage.

Creating SSL Decryption policies

Go to "Policies" --> "Decryption"

Create an "SSL Decrypt Exclude" Rule

It's best practive to have a "SSL Decrypt Exclude" rule in order to bypass SSL Decryption for Apps / URLs the PA cannot properly decrypt (see also Default Exceptions for details)

One thing you should have in this list is the "Unknown" category.

Create an "SSL Decrypt Exclude" Rule

Depending on your needs you next create a "SSL Decrypt Rule" specifying all URL categories you want to encrypt.

The configuration of the rules depend on your needs, but "Type" should always be "ssl-forward-proxy", the others are for SSH decryption and inboud decryption (i.e. the SSL Server is on your side)