The Scenario

Network Security Systems, including firewalls, can be configured to control (permit or deny) encrypted traffic, but cannot decipher the contents of the encrypted communication. The Secure Socket Layer (SSL) protocol and its predecessor, Transport Layer Security (TLS) protocol have become extremely popular choices for encrypting network communication, especially Internet web server traffic. Palo Alto Networks firewalls offer features to decrypt SSL/TLS traffic, providing increased visibility and threat protection. With the Palo Alto Networks Operating System (PAN-OS), firewalls can:

1. Decrypt Internet-bound web sessions – Palo Alto Networks firewalls use the "man-in-the-middle" technique to perform Internet-bound decryption, also known as "Forward Proxy Decryption."
2. Apply Application and Content Inspection – After traffic is decrypted, Palo Alto Networks firewalls can apply App-ID, and Content Inspection features to the decrypted or "Plaintext" traffic in real-time.
3. Perform Decryption Mirroring – Decrypted traffic can be forwarded to out-of-band security devices for further inspection and storage using port mirroring.

The Objective

In this lab, we will learn how to implement SSL/TLS Forward Proxy Decryption using Palo Alto Networks Next-Generation Firewalls in a Layer 3 deployment mode. We will also take advantage of the Decryption Port Mirror feature to allow for further analysis of decrypted data.

> 🌀 **TIP –** Decryption Mirroring is only available on the PA-7000, PA-5200, PA-5000, and PA-3000 series firewall appliances. In order to enable this feature, you must activate and install a free license and reboot the firewall. This procedure will be shown later in the lab.

The Tools

Accomplishing your objective will require the configuration of several objects and elements:

- ✓ SSL/TLS Forward Proxy Certificates
- ✓ Client Certificate Store(s)
- ✓ Decryption Profile
- ✓ Decryption Policy Rules
- ✓ Decryption Port Mirror License (free)
- ✓ Decryption Port Mirror Interface

The Setup

Our lab setup consists of a Palo Alto firewall running PANOS 8.0 and configured in Layer 3 mode with two network interfaces attached to separate security zones (Trust and Untrust), and one interface dedicated to decryption port mirroring. Internet-bound web traffic sourced by clients behind the Trust zone is decrypted, inspected, re-encrypted and forwarded to the ultimate destination web servers. Decrypted traffic is copied and forwarded to out-of-band security systems using the Decryption Mirror interface. (**Note:** It is assumed that the firewall is already configured in Layer 3 mode and passing network traffic).
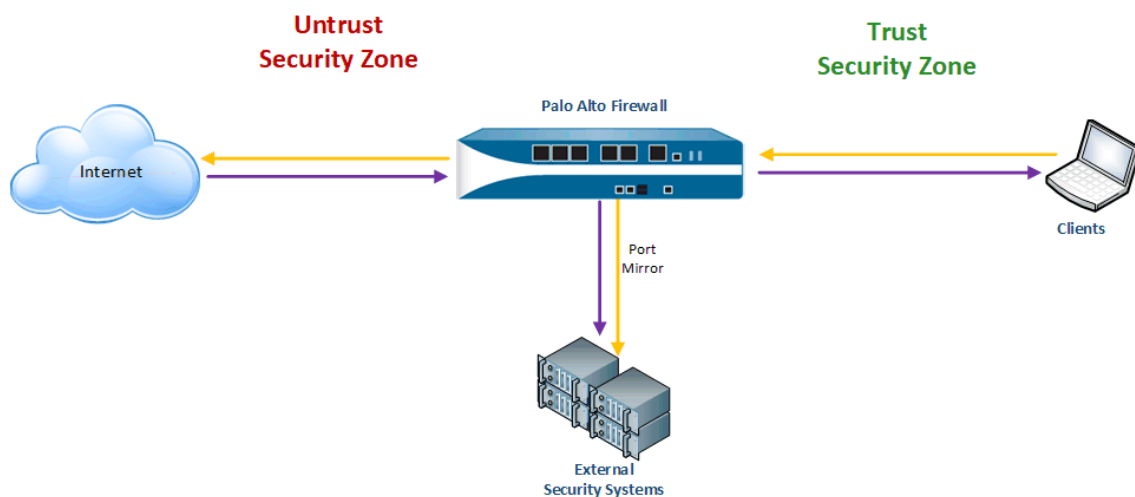

The Lab Configuration Steps

1. Certificate Management

a. Purpose
In order to perform SSL/TLS decryption, the firewall must issue certificates on the fly to clients on behalf of the web servers they are connecting to. These newly minted certificates must be signed by a Certificate Authority (CA) Public Key Infrastructure (PKI) certificate, where the firewall controls the public and private key pair. These PKI "signing" certificates can be self-signed or Intermediate CA certificates signed by an Enterprise Root CA. (**Note:** For the purposes of this lab, we will be using self-signed certificates).

b. Location
Certificates are configured in the *Device* tab under the *Certificate Management* group in the left menu and the *Device Certificates* sub tab.
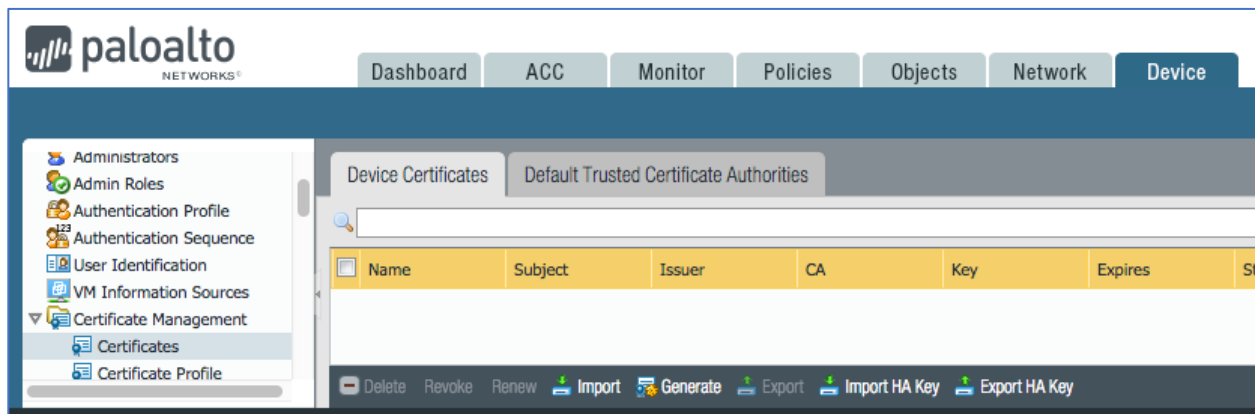
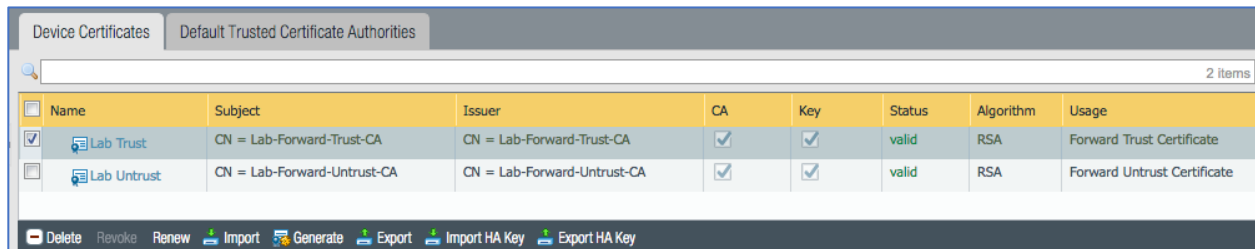c. **Creating a Forward Trust Self-Signed Certificate**

i. Click the *Generate* option at the bottom of the window
ii. **Certificate Name** = An internal object identifier for the certificate
iii. **Common Name** = A name that will appear as the Issuer of certificates sent to clients
iv. Check the *Certificate Authority* checkbox to create a self-signed certificate
v. (Optional) Adjust the *Cryptographic Settings* options as desired
vi. Click *Generate* to generate the certificate and private key
vii. Repeat the steps above to create an Untrust self-signed certificate

🍥 **TIP –** Creating separate Trust and Untrust certificates allows the firewall use different certificates when minting and signing certificates sent to clients. Setting the *Expiration* field of the Untrust certificate to 1 day will force this certificate to expire the next day, further increasing the chance of a client browser red flagging any untrusted certificate signed by it.

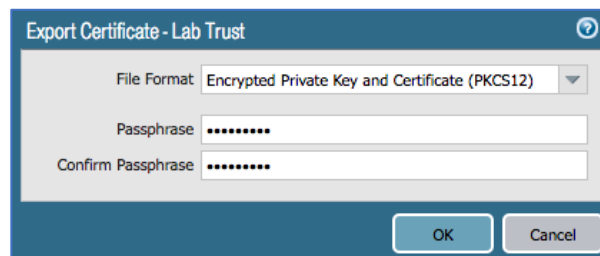d. **Configuring Forward Trust and Untrust Certificates**
   i. Once the certificates and private keys have been created click the name of each certificate to open up the **Certificate Information**
   ii. For the Trust certificate, select the **Forward Trust Certificate** checkbox
   iii. For the Untrust certificate, select the **Forward Untrust Certificate** checkbox
   iv. Click **OK** to save each setting
   v. Click **Commit** to commit your candidate configuration to running

e. **Exporting the Forward Trust Certificate**
   The Forward Trust certificate must be exported from the firewall in order for it to be installed on client devices in the next section.

   i. Select the checkbox to the left of the Forward Trust certificate and click the **Export** button at the bottom of the window
   ii. Choose a **File Format** of **Encrypted Private Key and Certificate (PKCS12)**
   iii. Enter a **Passphrase** and confirm
   iv. Click **OK** and save the file

2. Make Clients Trust the Certificate

a. **Purpose**
   Now that you have generated a Forward Trust certificate and private key to be used in signing trusted SSL/TLS connection certificates, you must ensure that the clients subject to decryption have the Forward Trust certificate installed and trusted by the OS, browser, and/or application certificate store. (**Note:** When

generating a using an Enterprise Root CA to sign a Forward Trust certificate generated by the firewall, this section is unnecessary if the Enterprise Root CA certificates are already installed and Trusted by clients).

> 🌐 **TIP –** Client device Operating Systems have a certificate store that is used by many applications loaded on the device, including default web browsers (IE, Edge, Safari), and some third-party web browsers (Chrome). However, other web browsers (Firefox) and applications that communicate using SSL/TLS have their own certificate stores that must be managed independently, if possible. If it is not possible to change an application certificate store, the traffic must be exempt from decryption in order for the application to continue to function.

    b. Location

       The location of each certificate store varies by OS and by application. In this lab, we will be installing the Forward Trust certificate in the Windows certificate store.

    c. Installing the Forward Trust Certificate in Windows

      i. On the Windows client system type *mmc* in the *Run* dialog box to open the Console

      ii. Click *File > Add/Remove Snap-in…* and add the *Certificates* snap-in for either the *User account*, the *Computer account*, or both, depending on which accounts you would like to have access to the Trust certificate

      iii. Click *Finish*

      iv. Navigate to *Trusted Root Certification Authorities* and expand it to select *Certificates*

      v. Click *Action > All Tasks > Import…*

      vi. Click *Next* and *Browse...* to the location where you saved the exported file

      vii. Select the file and click *Open* (**Note:** You may need to adjust your File Types next to the *File name:* field)

      viii. Click *Next* and enter the password you used when exporting the file

      ix. Click *Next* and make sure the *Trusted Root Certification Authorities* store is selected

      x. Click *Next* and then click *Finish* (**Note:** You may see a dialog box confirming the import of the certificate and private key. Choose *Yes*).

      xi. You should now see the Forward Trust certificate installed in the *Trusted Root Certification Authorities* store

3. Create a Decryption Profile

a. Purpose
A Decryption Profile allows you to perform checks and verification on sessions, certificates, and protocol versions, giving you granular access to control many scenarios the firewall could encounter when processing SSL/TLS traffic.
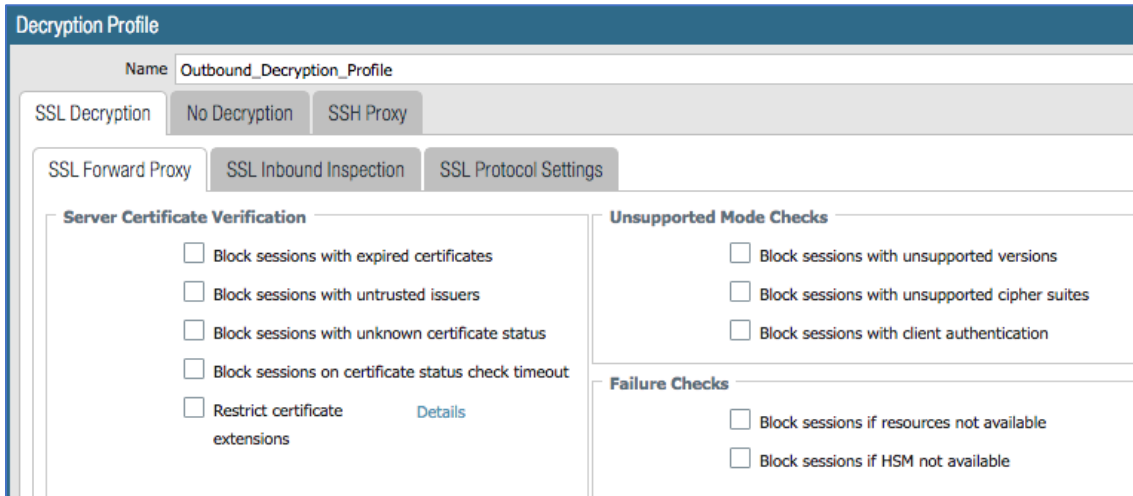
b. Location
Decryption Profiles are configured in the **Objects** tab under **Decryption Profile** in the left menu.

    c. **Building the Decryption Profile** (**Note:** You can utilize the default profile, clone the default profile, or create a new one from scratch)

> 🌅 **TIP –** Cloning the default or creating a new profile will allow you to make future changes to the profile without having to re-attach the profile to the decryption rules we will be creating later in the lab.

      i. **Name** = An object identifier that the firewall will use to reference in other parts of the configuration

      ii. **SSL Decryption** = Gives you three tabs of settings for controlling decrypted traffic. In this lab, we will be using the **SSL Forward Proxy** and **SSL Protocol Settings** sub-tabs.

        iv. **SSL Forward Proxy =** Allows you to perform Certificate Verification, Block Unsupported Sessions, and handle decryption failures. (**Note:** We will be using default settings for this lab).

        v. **SSL Protocol Settings =** Allows you to configure supported protocols and algorithms. (**Note:** We will be using default settings for this lab).

      iii. Click **Commit** to commit your candidate configuration to running

4. Create Decryption Policy Rules

    a. **Purpose**

Decryption Policy Rules allow the firewall to granularly match specific network traffic and apply SSL/TLS decryption. In addition, exclusion or override rules can be configured to exempt certain traffic from decryption. Decryption rules operate in a similar fashion to Security and NAT rules where the policy is evaluated from the top down and the action associated with the first rule matched is taken.

b. Location

Decryption Policy Rules are configured in the **Polices** tab under **Decryption** in the left menu.
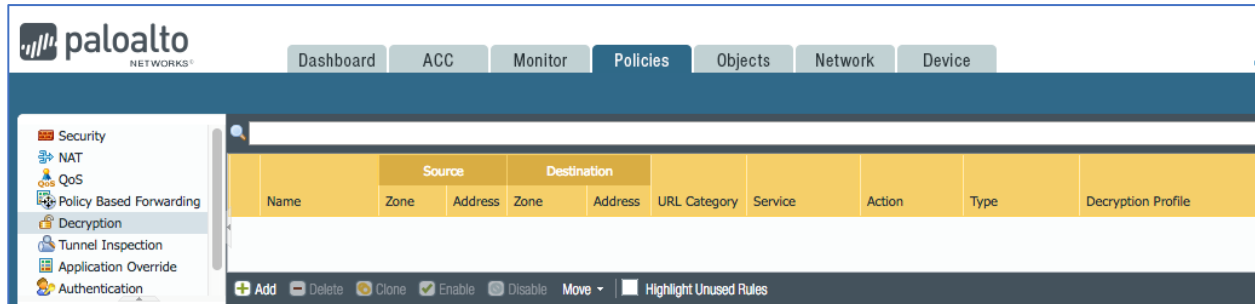
c. Building the Decryption Policy Rules

i. Click the **Add** button at the bottom of the window

ii. Enter a Name in the **Name** field

iii. Under the **Source** tab click **Add** under **Source Zone** and select zone **Trust**

iv. Under the **Destination** tab click **Add** under **Destination Zone** and select zone **Untrust**

v. Under the **Service/URL Category** tab click **Add** under **Service** and select service **service-https**

vi. Under the **Options** tab select an **Action** of **Decrypt**

vii. Set **Type** to **SSL Forward Proxy** and **Decryption Profile** to the profile created earlier in the lab

viii. Click **OK** to create your first decryption rule

ix. Click **Commit** to commit your candidate configuration to running
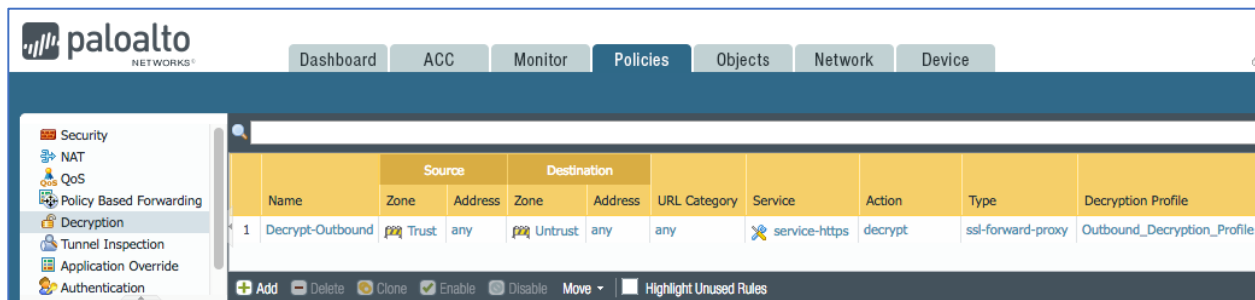
🌊 **TIP –** When implementing policies in a production environment, it is recommended that decryption be enabled slowly and methodically since it has the ability to break things. Using the decryption policy rules, you can start by matching a few source addresses, subnets, categories of URLs, etc…
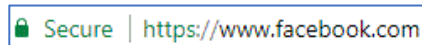
5. Test Decryption from the Client

a. Purpose

The initial configuration is done and the firewall is now configured to decrypt SSL/TLS traffic. Let's test to see if things are working as expected.
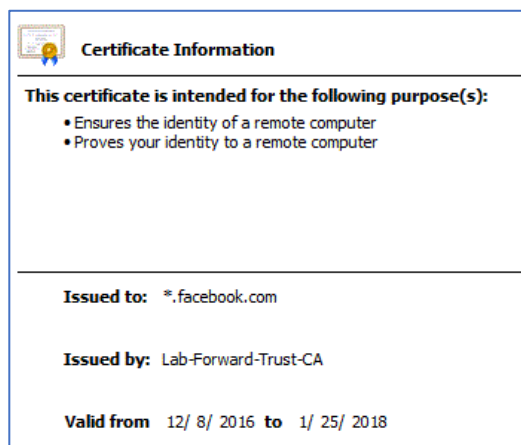
b. Testing from a Client Browser

i. From one of the client devices, open a browser (we are using Chrome 61 for this lab) and enter an HTTPS URL (e.g. https://www.facebook.com)

ii. You should see a Trusted green lock icon next to the URL

🔒 Secure | https://www.facebook.com

iii. Click the Settings icon to the right of the address box and choose ***More Tools > Developer Tools*** and select the ***Security*** tab (**Note:** You can also use the Ctrl+Shift+I keyboard shortcut)
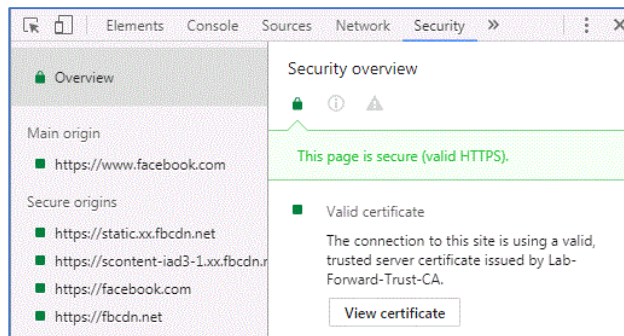
iv. Click the ***View Certificate*** button and verify that the Issuer of the certificate is the self-signed CA certificate you created earlier in the lab

Certificate Information

This certificate is intended for the following purpose(s):
- Ensures the identity of a remote computer
- Proves your identity to a remote computer

Issued to: *.facebook.com

Issued by: Lab-Forward-Trust-CA

Valid from 12/ 8/ 2016 **to** 1/ 25/ 2018

c. **Testing Threat Protection for Decrypted Content**

Now that the firewall is decrypting traffic, we can leverage the Next-Generation Threat Protection capabilities and apply them to decrypted traffic. (**Note:** For the purposes of this lab, it is assumed that the firewall has a valid Threat license installed and a Vulnerability profile has been configured to block vulnerabilities with severity *medium* and higher, or to use the default signature actions).

    i.   From one of the client devices, open a browser and enter the HTTPS URL to download the EICAR Test File (https://secure.eicar.org/eicar.com.txt)

    ii.   You should receive a *Virus/Spyware* Response Page from the firewall

---

Virus/Spyware Download Blocked

Download of the virus/spyware has been blocked in accordance with company policy. Please contact your system administrator if you believe this is in error.

File name: eicar.com.txt

🌊 **TIP –** If you have a WildFire profile configured to forward *PE* files to the Palo Alto Networks WildFire Public Cloud, you can test 0-Day Threat Detection over SSL/TLS Decrypted sessions using the https://wildfire.paloaltonetworks.com/publicapi/test/pe URL. To verify, check the *WildFire Submissions* firewall log within 5-10 minutes after the file download is complete.

6. Verify Traffic Logs on the Firewall

a. Purpose

It appears that the client browser is receiving the correct certificates from the firewall. Now let's verify the firewall is decrypting the traffic. Once sessions are decrypted, the firewall will have visibility into each session and will be able to apply App-ID and Threat Protection.

b. Location

Firewall Traffic and Threat Logs are located in the *Monitor* tab
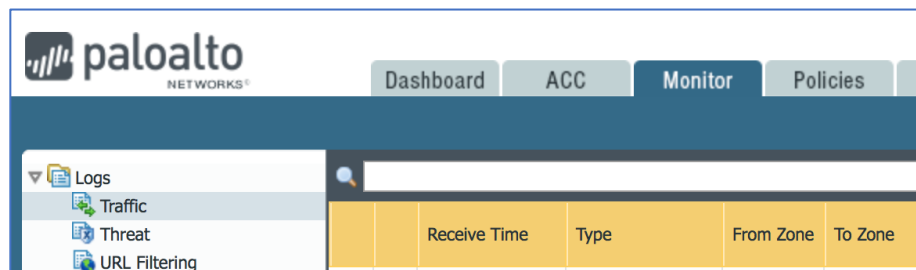
c. Verifying Decryption in Traffic Logs

  i.   Look through the traffic logs to find the session created earlier by going to the HTTPS URL on the client device. You can filter the traffic log to help you search. For example, *(app eq facebook-base)*.

 ii.   Click the Detailed Log View [icon] icon to the left of the traffic log

iii.   Verify that the *Decrypted* checkbox is checked for the session under *Flags*
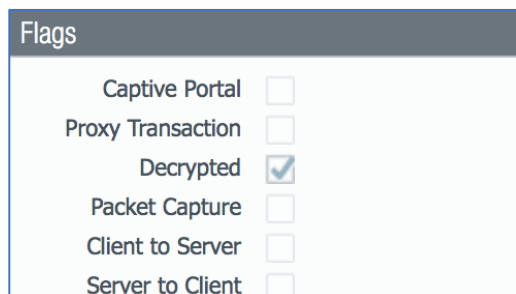
d. Verifying Threat Logs for Decrypted Content

  i.   Look through the threat logs to find the Vulnerability log associated with the denied session experienced earlier in the lab with a *File Name* of *eicar.com.txt*

	Type	Name	From Zone	To Zone	Attacker	Victim	To Port	Application	Action	Severity	File Name
	vulnerability	Eicar File Detected	Untrust	Trust	213.211.198.58	10.0.10.100	64291	web-browsing	reset-server	medium	eicar.com.txt
	vulnerability	Eicar File Detected	Untrust	Trust	213.211.198.58	10.0.10.100	56325	web-browsing	reset-server	medium	eicar.com.txt

7. Configuring Decryption Exemptions

a. Purpose

When implementing SSL/TLS Decryption in any network, there are always cases in which certain traffic needs to be exempt from decryption. Some examples include:
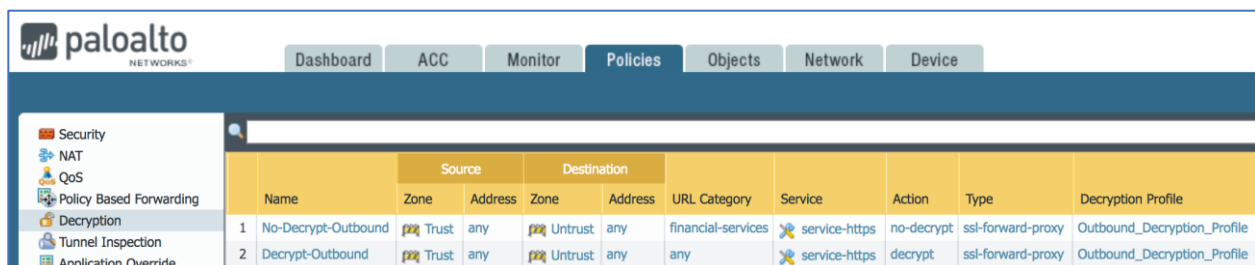
- Organization policy or privacy rules (e.g. Exempt Medical and Financial sites)
- Applications with their own certificate stores that cannot be modified
- Client certificate "mutual" authentication in which a client certificate is required by the server (e.g. SmartCard or CAC)
- Certificate Pinning in which the application expects a specific set of certificates from certain web servers

(**Note:** For the purposes of this lab we will be creating an exemption for Financial web sites).

> **TIP** – Starting with PANOS 8.0, the Palo Alto firewalls expose a list of URLs that are exempt from SSL/TLS Decryption by default. This list can be modified from the *Device* tab under *SSL Decryption Exclusion*. The pre-defined entries are updated through regular Content updates.

b. Building the Exemption Rules

i. Select the rule created earlier, click the *Clone* button at the bottom of the window, and choose *Before rule* for *Rule order*

ii. Edit the Name in the *Name* field

iii. Under the *Service/URL Category* tab click *Add* under *URL Category* and select the *financial-services* category

iv. Under the *Options* tab select an *Action* of *No Decrypt*

v. Set *Type* to *SSL Forward Proxy* and *Decryption Profile* to the profile created earlier in the lab (**Note:** You can use a decryption profile to validate certificates for sessions the firewall does not decrypt, or you can set this value to *None*).

vi. Click *OK* to create your first exemption rule

vii. Click *Commit* to commit your candidate configuration to running

8. Test the Exemption from the Client

    a. Purpose

    Now that we have a decryption exemption rule in place, let's test to make sure Financial sites are exempt from SSL/TLS decryption. (**Note:** In order to control SSL/TLS decryption and exemptions by URL Category, your firewall must be licensed with the URL Filtering license and receiving URL database updates).

    b. Testing from a Client Browser

      i. From one of the client devices, open a browser and enter an HTTPS URL for a Financial Service website (e.g. https://www.wellsfargo.com)

      ii. You should see a Trusted green lock icon next to the URL

      iii. Click the Settings icon to the right of the address box and choose ***More Tools > Developer Tools*** and select the ***Security*** tab (**Note:** You can also use the Ctrl+Shift+I keyboard shortcut)
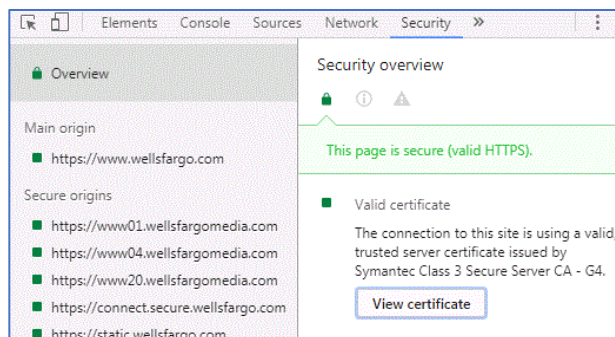
      iv. Click the ***View Certificate*** button and verify that the Issuer of the certificate is the original public Internet Root CA (e.g. Symantec)
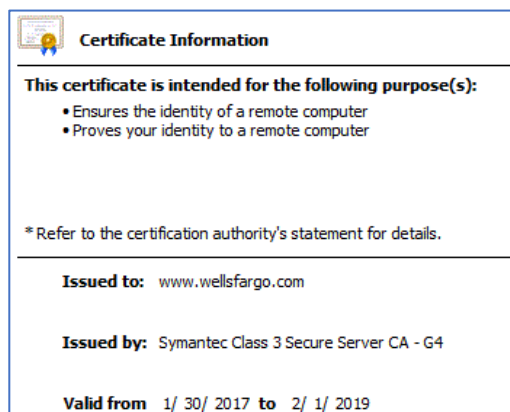
9. Set up Decryption Port Mirroring

a. Purpose

Decryption Port Mirroring allows the firewall to make a copy of the Plaintext traffic and forward it to external security and logging systems. You need to activate a free license from the Palo Alto Networks Support Portal to enable this feature. (**Note:** You will need to Reboot the firewall once the license has been installed in order to activate the feature).

b. Location

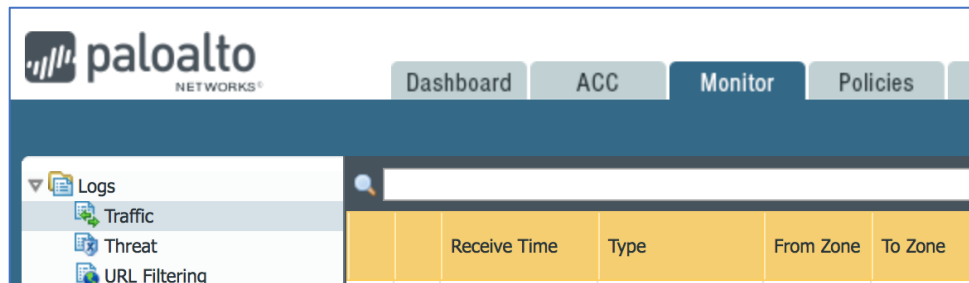The Decryption Port Mirroring License can be activated through the Palo Alto Networks Support Portal at https://support.paloaltonetworks.com. You will then need to install the license from the *Device* tab under *Licenses* in the left menu.

c. Activating the Decryption Port Mirror License

  i.   Log in to your support account at https://support.paloaltonetworks.com
  ii.   Navigate to the *Assets* tab locate your firewall
  iii. Click the *Actions* icon to the right of the *Licenses* listed

Model Name	Device Name	Group	License	Actions
PAN-PA-5050-NFR	Demo PA-5050 1		Renewal for NFRB Threat Prevention ⊻ BrightCloud URL Filtering ⊻ PAN-DB URL Filtering ⊻ GlobalProtect Gateway ⊻ GlobalProtect Portal ⊻ Standard Support WildFire License ⊻	✏️

  iv.   Select the option to *Activate Feature License* and check the *Decryption Port Mirror* checkbox

ACTIVATE LICENSES
○ Activate Auth-Code
● Activate Feature License

AVAILABLE FEATURE LICENSES
☑ Decryption Port Mirror
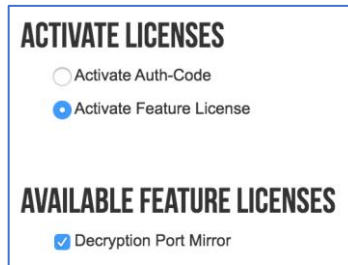
v.   Click **Agree and Submit**

d. **Installing the Decryption Port Mirror License**

i.   From the firewall GUI, navigate to **Device > Licenses**
ii.   Click the **Retrieve license keys from license server** link
iii. Verify the **Decryption Port Mirror** license is listed
iv.   Reboot the firewall by clicking the **Reboot Device to activate** link and clicking **Yes**

Decryption Port Mirror

Date Issued   October 04, 2017
Date Expires Never
Description   Decryption Port Mirror
Active   No (Reboot device to activate)

10. Configure the Decryption Mirror Interface

a. **Purpose**
Decryption Port Mirroring requires a dedicated an interface on the firewall. Once the Decryption Port Mirroring license is installed in the firewall you will have the option to configure interfaces with type **Decrypt Mirror**.

b. **Location**
The Decryption Mirror interface is configured in the **Network** tab under **Interfaces.** The interface is then selected through the **Decryption Profile** located in the **Objects** tab under **Decryption Profile.**
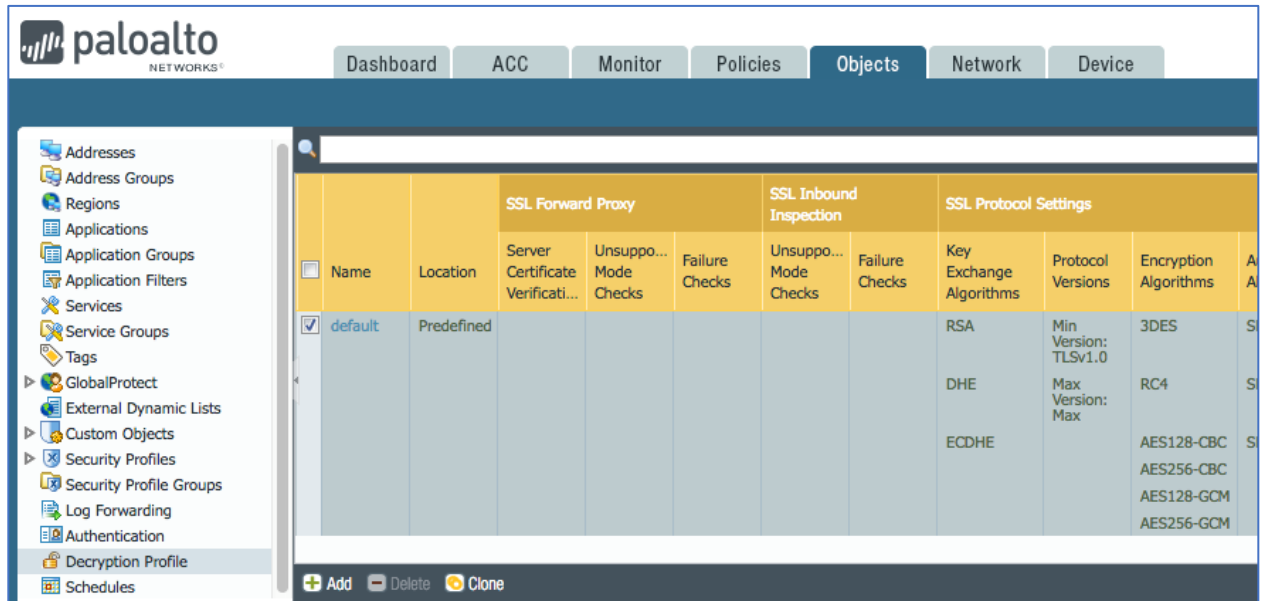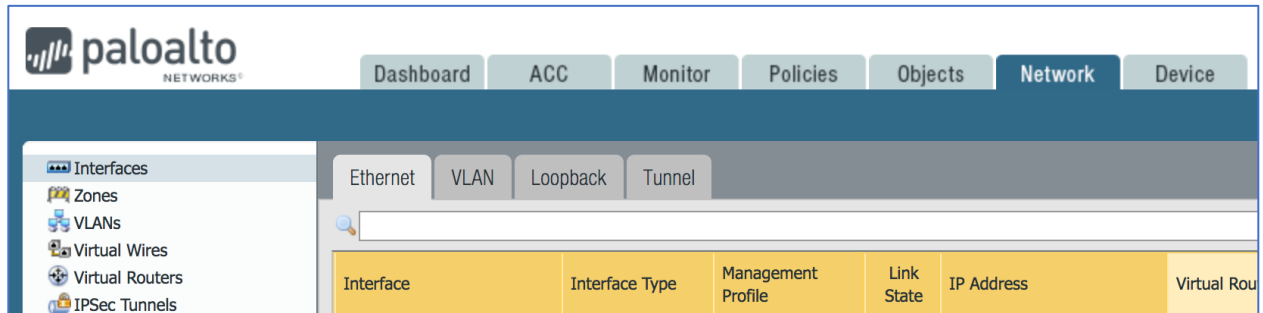
c. Configuring the Decryption Mirror Interface

    i.   Click the name of the interface you wish to configure for Decryption Mirroring

    ii.   Set the *Interface Type* to *Decrypt Mirror*

d. Selecting the Decryption Mirror Interface

 i. Navigate to the *Objects* tab and choose the *Decryption Profile* option from the left menu

 ii. Click the name of the profile you created earlier in the lab
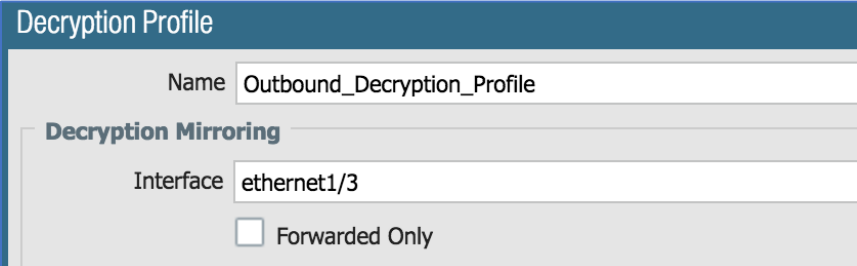
 iii. Under *Decryption Mirroring* choose the *Interface* that you configured as type *Decrypt Mirror*

 iv. Uncheck the *Forward Only* checkbox. (**Note:** This option will allow all traffic to be mirrored through the Decrypt Mirror interface when unchecked, including traffic for sessions that are denied by the firewall).

Decryption Profile

Name | Outbound_Decryption_Profile
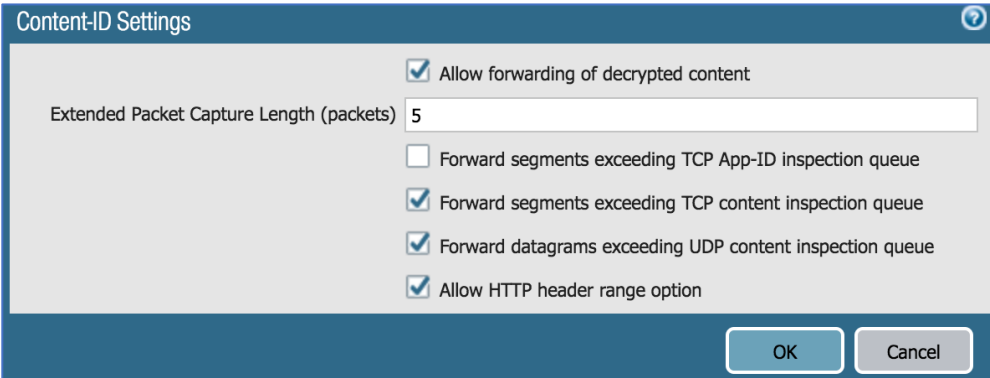
Decryption Mirroring

Interface | ethernet1/3

☐ Forwarded Only

e. Activate Decrypted Traffic Forwarding

 i. Navigate to the *Device* tab, choose the *Setup* option from the left menu, and select the *Content-ID* sub tab

 ii. Click the "gear" icon in the upper-right corner of the *Content-ID Settings* widget to edit the settings

 iii. Check the *Allow forwarding of decrypted content* checkbox and click *OK*

 iv. Click *Commit* to commit your candidate configuration to running

Content-ID Settings

☑ Allow forwarding of decrypted content

Extended Packet Capture Length (packets) | 5

☐ Forward segments exceeding TCP App-ID inspection queue

☑ Forward segments exceeding TCP content inspection queue

☑ Forward datagrams exceeding UDP content inspection queue

☑ Allow HTTP header range option

OK | Cancel

11. Verify Forwarding of Decrypted Traffic

a. **Purpose**

Now that we have Decryption Port Mirroring configured and enabled on the firewall we can verify that the external systems are receiving the out-of-band, Plaintext network traffic. (**Note:** For the purposes of this lab, we have the Decrypt Mirror interface connected to a workstation running packet capture software).

b. **Verify Decryption Port Mirroring**

i. Using the client test device, generate some traffic to a few HTTPS websites that are not exempt from decryption (e.g. https://www.ssllabs.com)

ii. Capture traffic on the Decryption Port Mirror workstation using a packet capture tool (e.g. WireShark)

iii. Verify the captured traffic is Plaintext:

iv. Destination port is **443**

v. We can see the **HTTP** protocol methods, like **GET**

vi. Payloads are in Plaintext, not encrypted

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.10.100	64.41.200.100	TCP	62	56349 → 443 [SYN] Seq=0 Win=16383 Len=0 MSS=1360 WS=4
2	0.000000	64.41.200.100	10.0.10.100	TCP	62	443 → 56349 [SYN, ACK] Seq=0 Ack=1 Win=16383 Len=0 MSS=1360 WS=4
3	0.000000	10.0.10.100	64.41.200.100	TCP	60	56349 → 443 [ACK] Seq=1 Ack=1 Win=65532 Len=0
4	0.000000	10.0.10.100	64.41.200.100	HTTP	508	GET / HTTP/1.1
5	0.136000	64.41.200.100	10.0.10.100	TCP	497	[TCP segment of a reassembled PDU]
6	0.136000	64.41.200.100	10.0.10.100	TCP	1414	[TCP segment of a reassembled PDU]
7	0.136000	10.0.10.100	64.41.200.100	TCP	60	56349 → 443 [ACK] Seq=455 Ack=1804 Win=65532 Len=0
8	0.136000	64.41.200.100	10.0.10.100	TCP	699	[TCP segment of a reassembled PDU]
9	0.136000	64.41.200.100	10.0.10.100	HTTP	62	HTTP/1.1 200 OK (text/html)
10	0.136000	10.0.10.100	64.41.200.100	TCP	60	56349 → 443 [ACK] Seq=455 Ack=2457 Win=65532 Len=0
11	0.172000	10.0.10.100	64.41.200.100	HTTP	533	GET /news-iframe.html HTTP/1.1
12	0.260000	64.41.200.100	10.0.10.100	TCP	583	[TCP segment of a reassembled PDU]
13	0.260000	64.41.200.100	10.0.10.100	TCP	676	[TCP segment of a reassembled PDU]

```
▶ Frame 4: 508 bytes on wire (4064 bits), 508 bytes captured (4064 bits)
▶ Ethernet II, Src: Vmware_92:80:08 (00:50:56:92:80:08), Dst: PaloAlto_00:01:11 (00:1b:17:00:01:11)
▶ Internet Protocol Version 4, Src: 10.0.10.100, Dst: 64.41.200.100
▶ Transmission Control Protocol, Src Port: 56349, Dst Port: 443, Seq: 1, Ack: 1, Len: 454
▼ Hypertext Transfer Protocol
  ▶ [Expert Info (Warning/Security): Unencrypted HTTP protocol detected over encrypted port, could indicate a dangerous misconfiguration.]
  ▶ GET / HTTP/1.1\r\n
    Host: www.ssllabs.com\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate, br\r\n

0000  00 1b 17 00 01 11 00 50  56 92 80 08 08 00 45 00   .......P V.....E.
0010  01 ee 8f 42 00 00 40 06  cc d6 0a 00 0a 64 40 29   ...B..@. .....d@)
0020  c8 64 dc 1d 01 bb 72 03  84 e4 64 e2 08 5a 55 18   .d....r. ..d..ZU.
0030  3f ff e9 05 77 db 47 45  54 20 2f 20 48 54 54 50   ?...w.GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 77 77 77 2e   /1.1..Ho st: www.
0050  73 73 6c 6c 61 62 73 2e  63 6f 6d 0d 0a 43 6f 6e   ssllabs. com..Con
```

Next Steps

If you want to test this on your own and do not have access to a lab environment to do so, you have a couple options:

a. Contact your Sun Management Account Rep to get pricing on a lab bundle. The newly released PA-220 and VM-50 appliances are excellent platforms for testing things such as this and there are specific part numbers for lab equipment that are more heavily discounted than the same appliance for use in production.

   If you are unsure who your Account Rep is or do not have one yet, you can reach out to **sales@sunmanagement.net** for assistance.

b. Reach out through the free Fuel Users Group (www.fuelusersgroup.org) which at the time this lab is being written is offering limited free access to a virtual lab environment, which they refer to as their "Virtual Test Lab," in which you can practice the steps outlined above. (Note: The Fuel Users Group may alter or discontinue offering their "Virtual Test Lab" at any time)

If you feel Sun Management brings value to you and your organization with these labs, please keep us in mind for other network and network security related requirements. We are here to help you. Thank you for your business.

Please direct any questions/comments/feedback on this lab exercise to: ***education@sunmanagement.net***

Lab Author: Mike Bermudez
             Sr. Network Security Engineer, PCNSE

Last Modified: Oct 6, 2017