Site-To-Site VPN

*Self-Signed Certificates*

Adv Cisco Cybersecurity – Lab 9

Jason Liu

Cybersecurity – Mr. Mason & Mr. Hansen

Period 5

*Lab 9: Palo Alto Site-To-Site with Self-Signed Certificates*

**Purpose**

The objective of this lab is to initialize and to set up an encrypting site-to-site connection between two client networks using Palo Alto, PA-410 configurations. Then, to monitor the session via an intervening switch to inspect the packet protocols for proof of encryption. This is the same as Lab 8. However, authentication should be done through a common certificate rather than a single pre-shared key.

**Background Information**

Background information of *Lab 8: Site-To-Site VPN with Pre-Shared Keys* is cited for the majority of the reference. Specific details relating to the nuanced authentication method of self-signed certificates are mentioned first.

Using digital certificates instead of pre-shared keys for IKE authentication, IPSec tunnels can be built without the need of remembering the manual input of a pre-shared key, which greatly reduces security risk. It also allows both static and dynamic customer gateway IP addresses.

The certificate, when exported/imported, can be either PEM or PKCS12 as the file format. PEM stands for Base64 Encoded Certificate and contains the certificate without the key and is written in cleartext. PKCS12 stands for Encrypted Private Key and Certificate, which acts both the certificate and key. When the certificate (generated by the root or subordinate CA) is exported, it defaults to the PEM file format. A Key File must also be imported for PEM. Palo Alto can allow PEM to be exported with the key file, however. This allows us to continue using PEM for this lab. A passphrase is also needed for the peer to use the PEM or PKCS12 file. The passphrase is not the key.

For certificates, it is still necessary for Local and Peer identification. This time, instead of statically assigned local and peer addresses, the Distinguished Name (Subject) can be used by referring to the name of the certificate.

During the IKE gateway configuration, use the shared self-signed certificate on both peers as the form of authentication. Once the firewalls have matching information regarding the certificates, the tunnel can be authenticated and then established.

Sources - <https://docs.paloaltonetworks.com/pan-os/9-1/pan-os-admin/vpns/set-up-site-to-site-vpn/set-up-an-ike-gateway/import-a-certificate-for-ikev2-gateway-authentication>.

<https://docs.paloaltonetworks.com/pan-os/9-1/pan-os-admin/vpns/set-up-site-to-site-vpn/set-up-an-ike-gateway#id47a6f121-466d-48fa-96f6-b122cd225c06_id910ed321-65d9-4eba-8cb2-010f2e64cafa>.

<https://repost.aws/knowledge-center/vpn-certificate-based-site-to-site>.

THE FOLLOWING IS CITED FROM *LAB 8 – SITE-TO-SITE VPN WITH PRE-SHARED KEYS* FOR DOCUMENTATION REFERENCE

A site-to-site Virtual Private Network (VPN) configuration connects two entire networks through an encrypted tunnel. One such tunnel is the IPSec tunnel. This provides secure connection through mediums that may be less secure, have different addressing schemes, or different IP protocols entirely.

In terms of IP addressing, the IP header and payload is embedded into another IP payload under a header specifically meant for the IPSec tunnel. The source IP address in the new header is of the local VPN peer’s address, and the destination IP is the opposing VPN peer’s address, where the tunnel ends. The addresses of the ends of the tunnel are of the same subnet, which is reflected in the corresponding topology. When the packet reaches the opposite peer’s firewall after the tunnel, the outer header is removed and the original packet is sent to its destination. This is reflected in the tunnel interface configurations. The encrypted information in the TCP/IP packet would appear as ESP throughout the whole tunnel, which is the protocol type seen in the monitored session on WireShark.

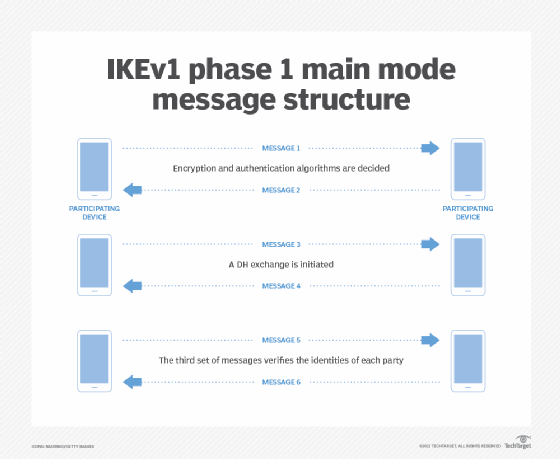


<https://docs.paloaltonetworks.com/pan-os/9-1/pan-os-admin/vpns/site-to-site-vpn-overview>

For example, when one PC from one LAN attempts to reach the other LAN via the tunnel, VPN Peer A initiates a connection request to VPN Peer B. If the associated security policy allows it, IKE establishes the connection over the said IPSec tunnel.

Internet Protocol Security (IPSec) is the name for a suite of protocols that help secure connections of the internet. Each with a different purpose, one main protocol is the Internet Key Exchange (IKE) used everywhere in this lab. In IPSec, IKE automatically negotiates Security Associations (SA), which acts as security policies for establishing a connection. Its algorithms and mutually agreed upon keys, which may include pre-shared keys, are used when the VPN’s are attempting to form a connection. Specifically, IKE uses X.509 public key infrastructure (PKI) certificates for authentication and a Diffie-Hellman key exchange protocol to establish a shared secret session. The protocol uses User Datagram Protocol packets (UDP) to create the SA, needing 4-packets across 2-3 messages.

The IKE has two phases, which is represented by the IKE crypto profile and IPSec crypto profile. Phase 1 operates in main or aggressive mode, sending 2 messages of exchanging encryption and authentication algorithms, 2 messages of a Diffie-Hellman key exchange (where both provide a random number), and 2 more messages to verify and authenticate the communication that will occur in phase 2. Aggressive mode disregards the third set of messages, losing out on certain parts of the encryption. This establishes a connection and a secure channel.

 Phase 2 negotiates the SA to ensure the data is secure while traveling through the IPSec channel created in phase 1. Both firewalls exchange information to determine security parameters for the SA. This means you would have at least two SAs that are unidirectional, one for each direction. Phase 2 operates in the “quick” mode, which provide proxy IDs, perfect forward secrecy, and replay protection. Proxy IDs are used as a form of identification between the two firewalls, and the forwarding ensures keys that are delivered to be unique and independent.

This image above describes the IKE messaging process, found on one of the research webpages: <https://www.techtarget.com/searchsecurity/definition/Internet-Key-Exchange>.

In general, IKE is often used in the IPSec for its practical applications such as automatic negotiation and authentication, replay protection, changing encryption keys mid-session, fast NAT traversal and calculating NAT connections, preventing DoS attacks, and having auto restoration as a form of fault tolerance.

Once IKE is established for the tunnel, the packet will successfully be delivered to the network at the other end of the tunnel.

**Lab Summary**

The Lab Summary of *Lab 8: Site-To-Site VPN with Pre-Shared Keys* is cited for the majority of this overview. Specific details relating to the nuanced authentication method of self-signed certificates are mentioned first.

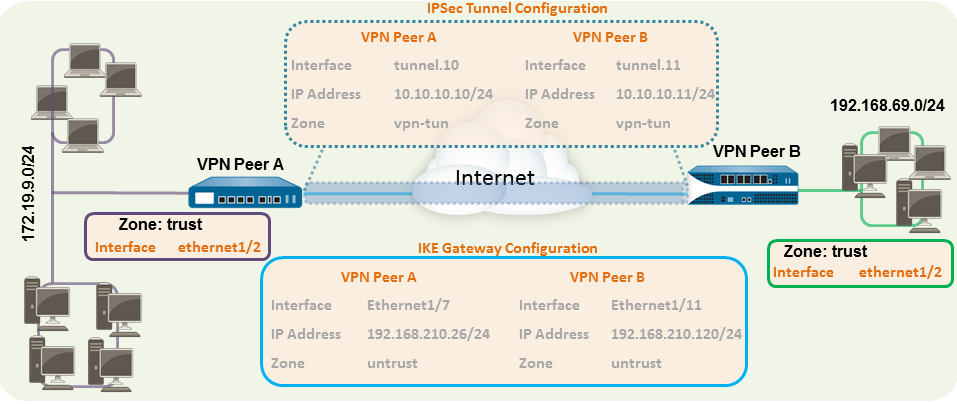
For the certificates, create a Root Certificate Authority (CA) under Device > Certificates. Under that authority, generate a certificate. This will be the one certificate used for the authentication. Export that certificate to be able to send it to the VPN peer. A passphrase is needed for the imported device to use the certificate. Create a certificate profile that uses the generated certificate.

On the peer firewall, import that certificate and enter the created passphrase. This will provide the certificate information. Create a certificate profile that uses the imported certificate.

Once the certificate is created, ensure that the IKE gateway (that the tunnel is using) is set to “certificate” mode and the generated certificate is assigned. Identify the Peer IDs using Distinguished Names, which are the names of the certificates. If the certificates were properly generated and assigned, the authentication part of this lab is functioning and the rest of the configuration is the same.

THE FOLLOWING IS CITED FROM *LAB 8 – SITE-TO-SITE VPN WITH PRE-SHARED KEYS* FOR DOCUMENTATION REFERENCE

This lab followed an adaption of the quick-configuration setup on Palo Alto’s PAN-OS Administrator’s Guide: <https://docs.paloaltonetworks.com/pan-os/9-1/pan-os-admin/vpns/site-to-site-vpn-quick-configs/site-to-site-vpn-with-static-routing>. Noticing that some addresses pointed incorrectly in this official document, corrections were made. Interface numbers were also changed to fit physical accessibility for the lab. The following topology was used:



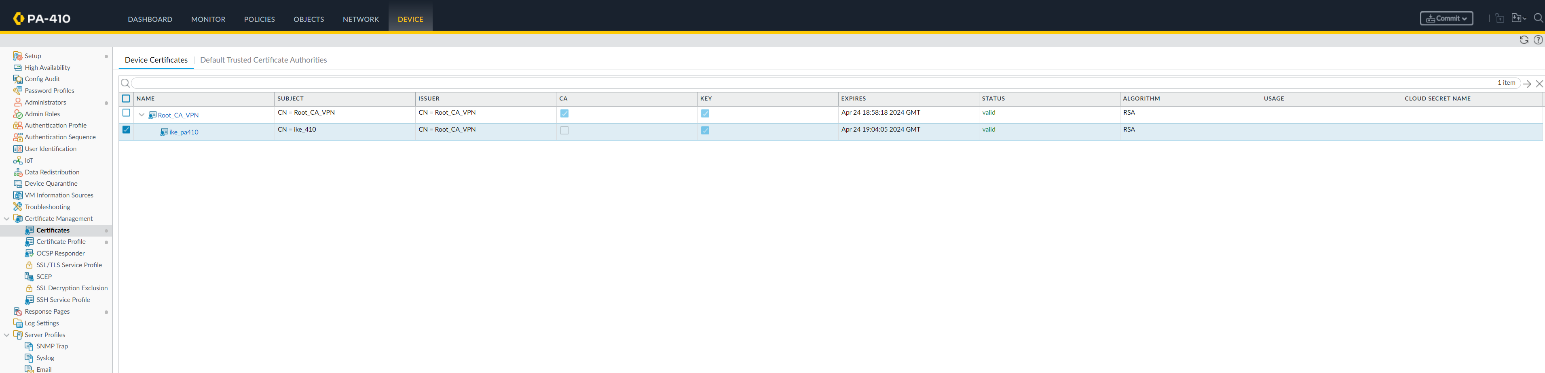
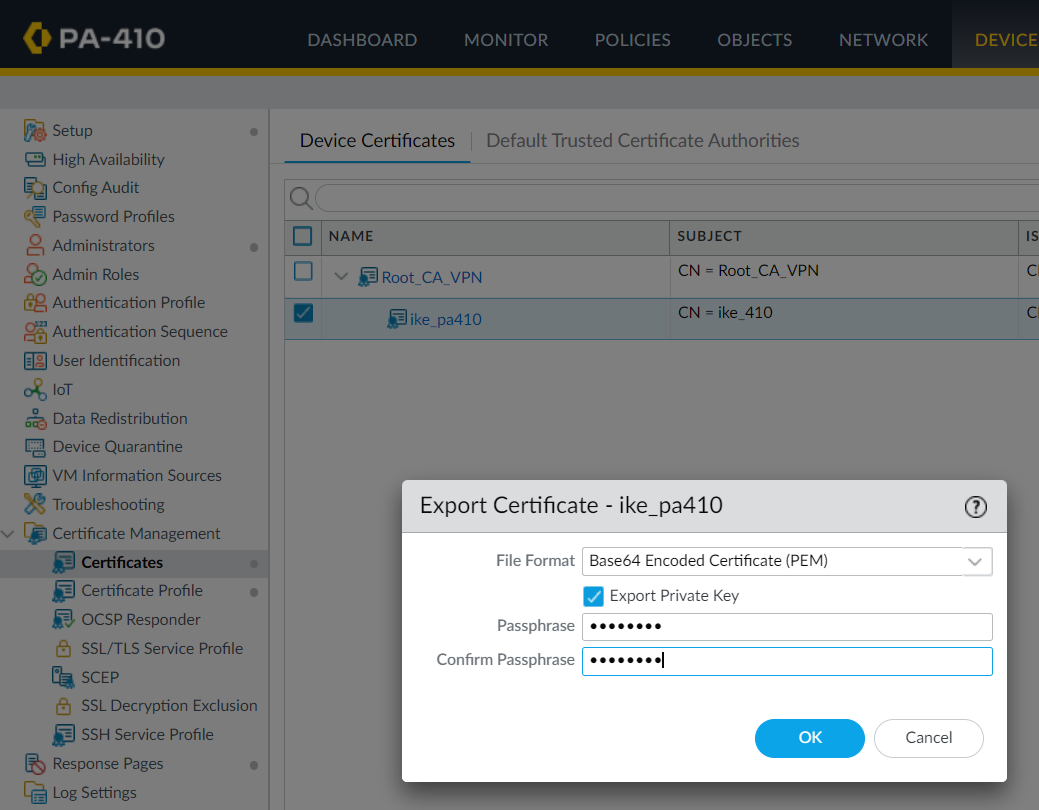
There are 3 main steps to the setup of this lab.

First, layer 3 ethernet interfaces have to be made on the firewall that correspond to their inside or outside security zones. These could also be named trust and untrust, respectively. One zone and one interface are unique, called the tunnel interface and the VPN tunnel zone, which acts as a virtual interface designed for the tunnel’s communication. Security policies should also be modified to allow traffic between the zones.

Second, authentication through IKE and IPSec Crypto files must be configured for the IKE gateway. The gateway and the virtual router were both configured/routed statically to access the subnet that is at the other end of the tunnel. This is where the certificates should be used as authentication. The route on the virtual tunnel should be created so that it uses tunnel IPs, not the physical interfaces.

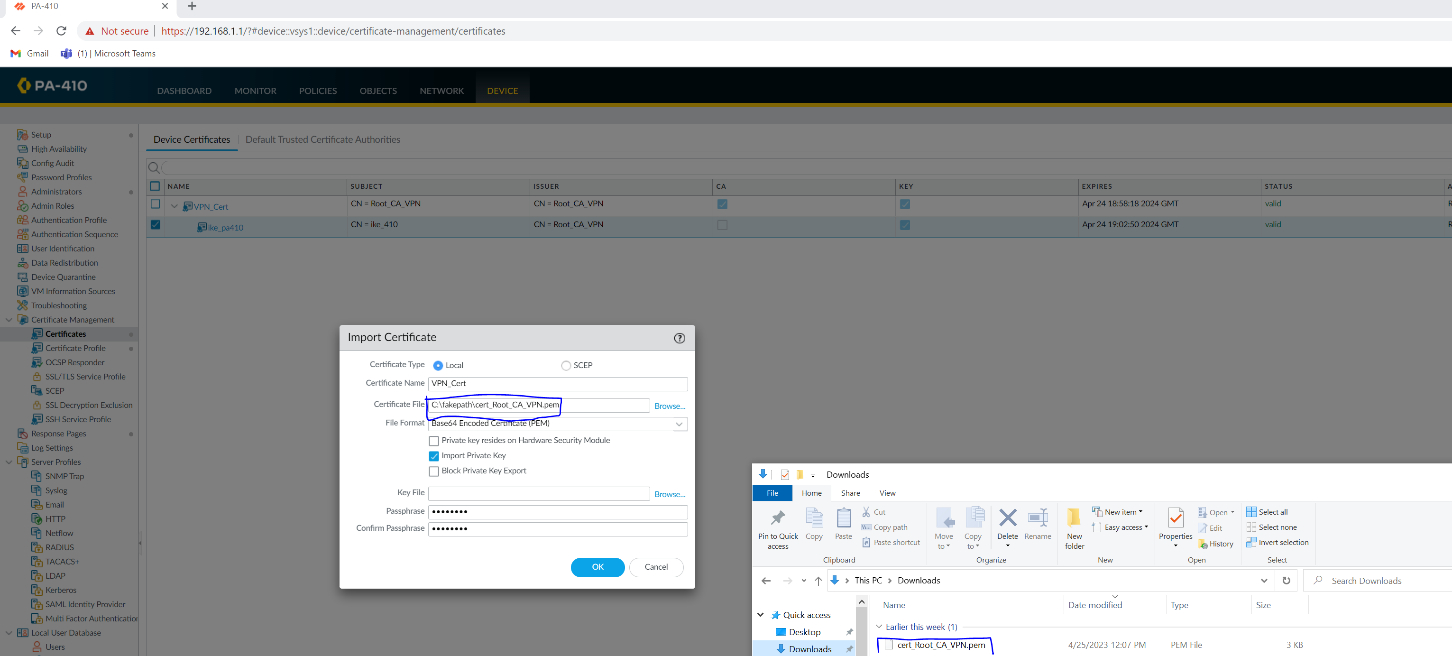
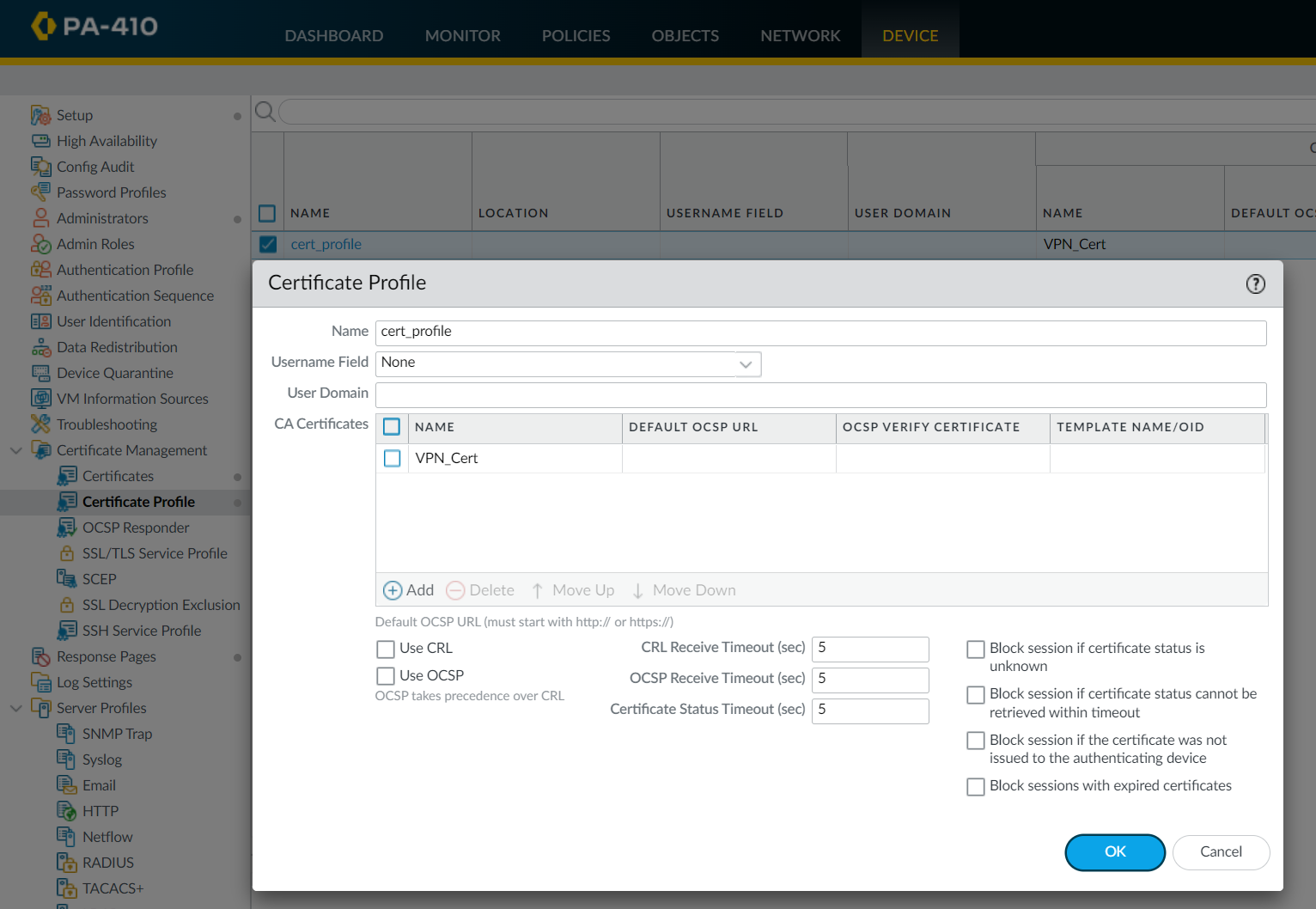
Lastly, the IPSec tunnel uses the IKE gateway and corresponding tunnel interfaces to fully establish the link between the two firewalls. Tunnel Monitoring on the Palo Alto console and the Monitor Session on an inconsequential Cisco Switch in the middle were used to debug, troubleshoot, and capture proof of encrypted packets being sent via the established IPSec tunnel. The command should appear as follows into the switch’s console - monitor session [session #] [source/destination] interfaces [interface].

**Procedure**

****

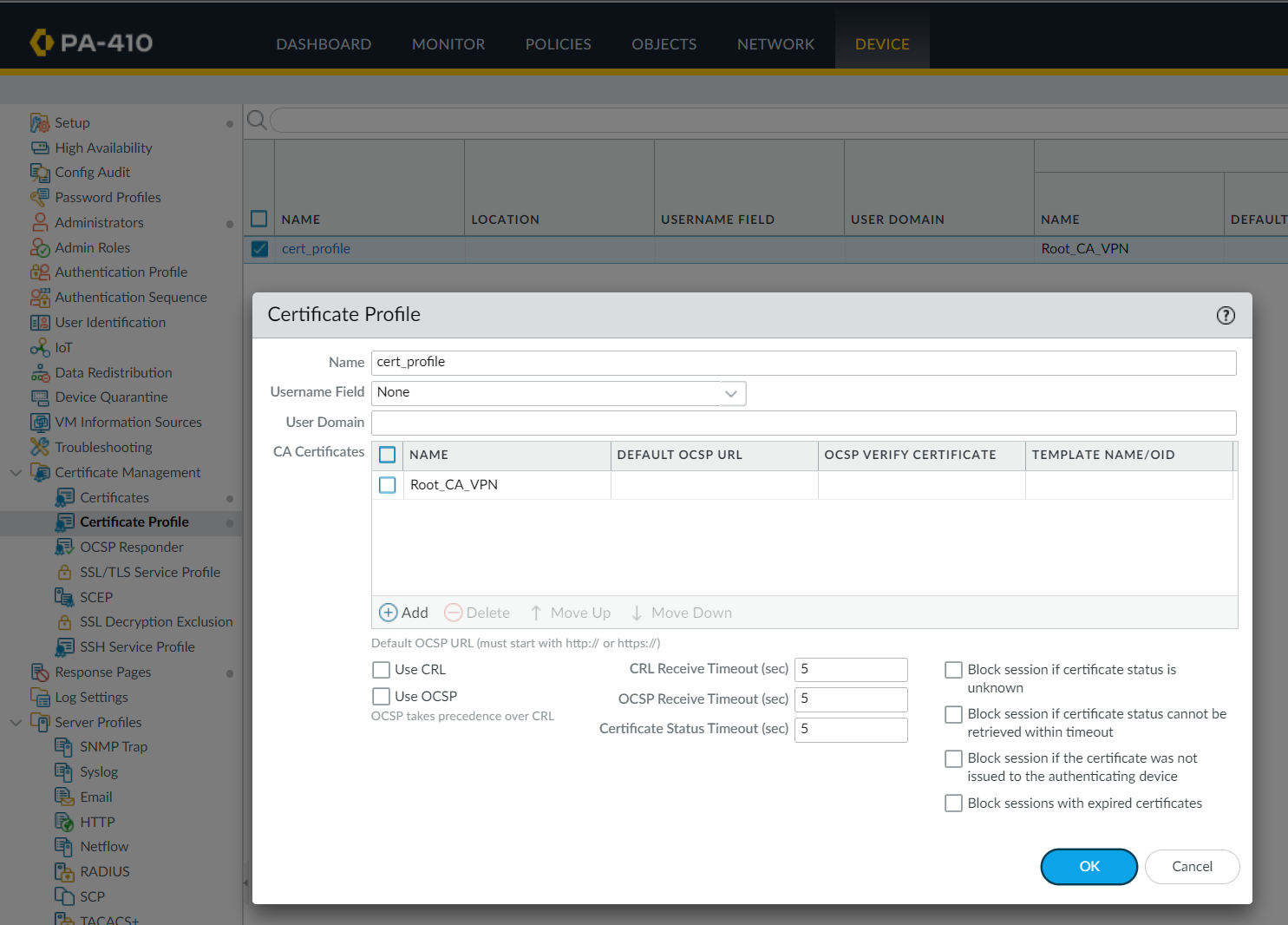
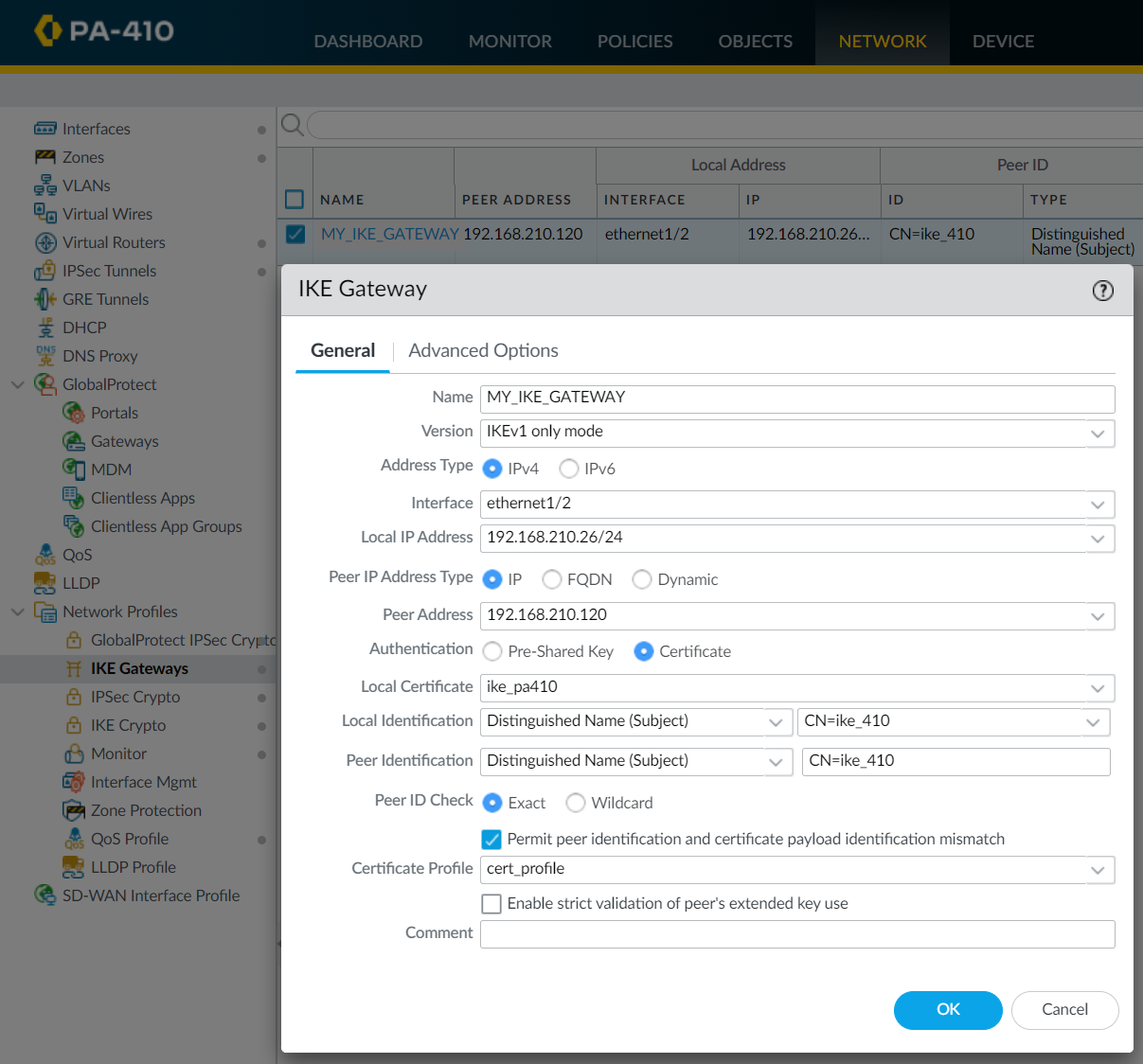
Export the certificate with the private key and enter a passphrase for when the peer firewall needs to import it.

Create a Root Certificate Authority and have it generate a certificate.

****

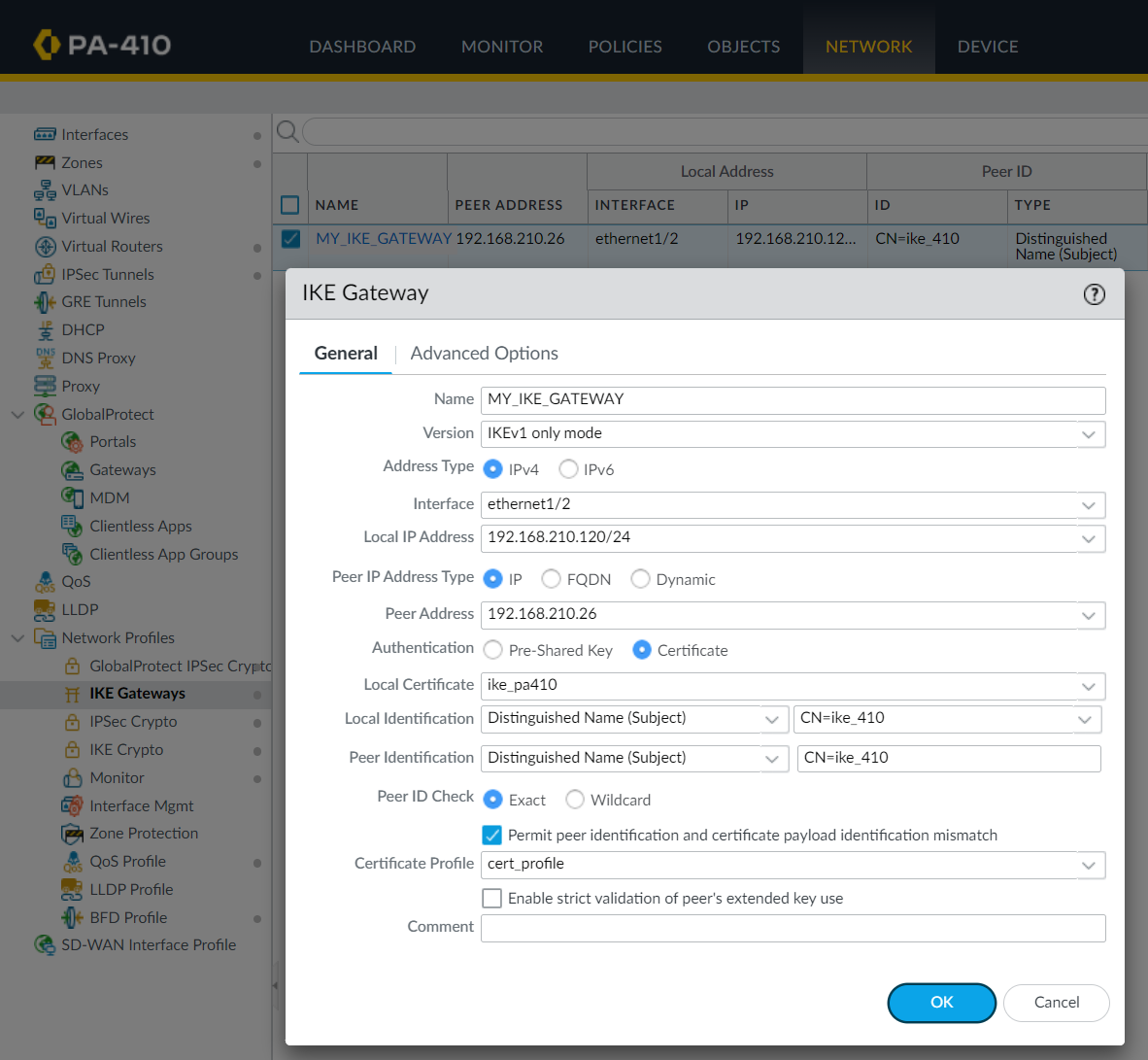
The Certificate Authority is named VPN\_Cert on Peer A, the imported firewall. Create the certificate profile that uses the created CA.

Import the certificate on the other firewall. This image was on Peer A, as Peer B was the root. The naming for the certificate authority does not need to be congruent.

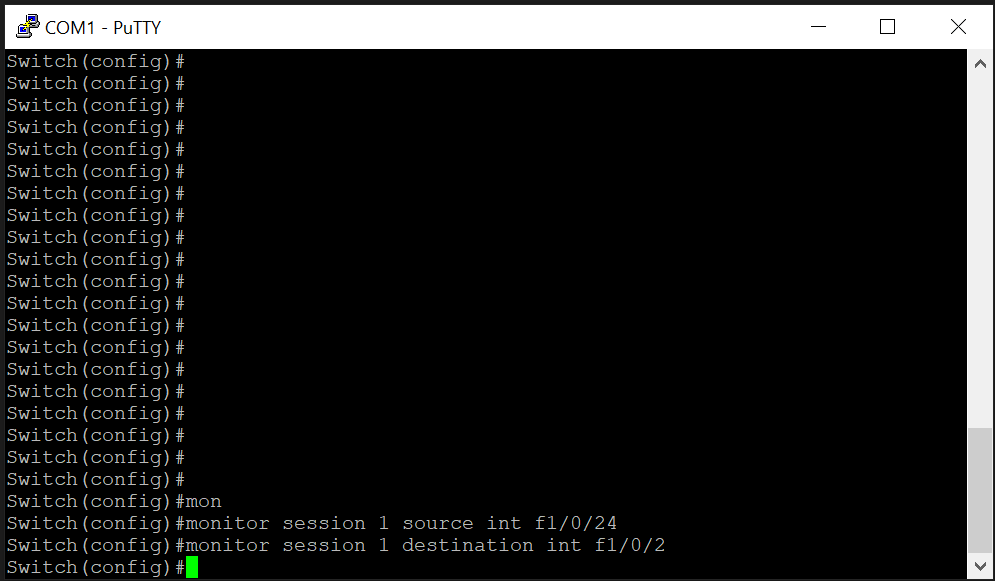
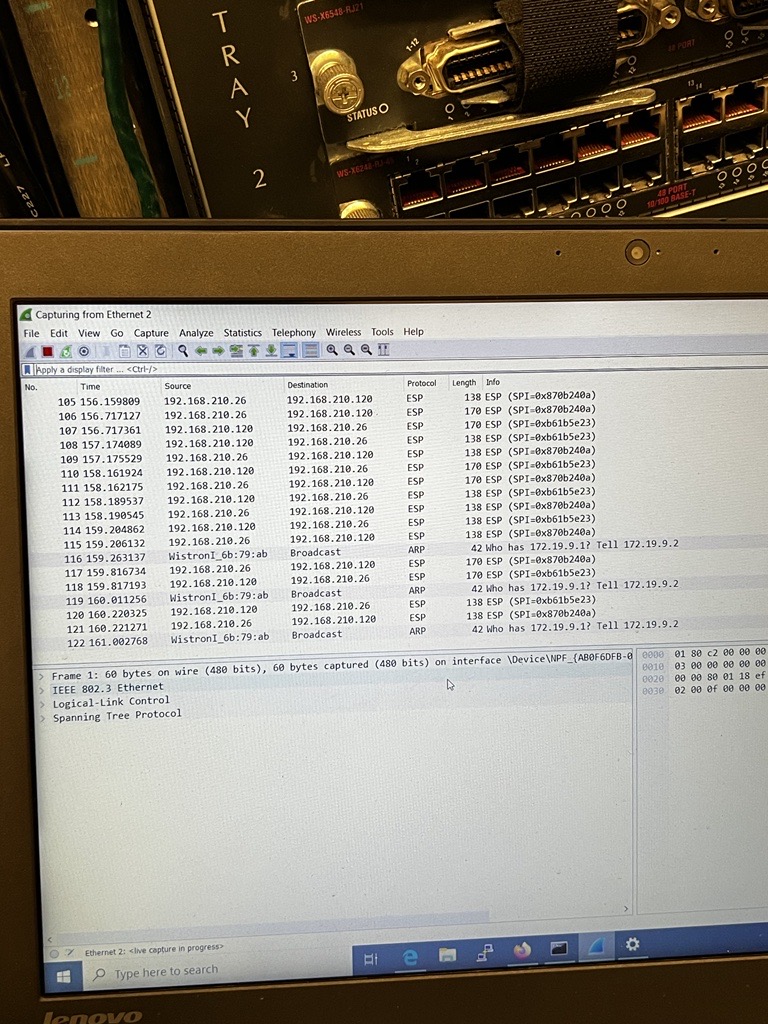
****

Edit the IKE gateway to use certificates as the form of authentication. Select the local certificate and enter the local and peer IDs via Distinguished Name (Subject).

The Certificate Authority is named Root\_CA\_VPN on Peer B, the root firewall. Create the certificate profile that uses the created CA.

****

Edit the IKE gateway on the other firewall as well. Notice the local and peer addresses are reversed compared to the previous image.

****

Enable monitor session using these commands on the intervening switch. Plugged into the destination interface, the WireShark captures the copied packets and shows that the sent packets from one of the firewalls are encrypted, shown by the ESP tag as the protocol.

**Problems**

This lab was completely timely, backing off the same IP scheme as *Lab 8: Site-to-Site VPN with pre-shared keys*. The only change was doing signed certificates as the form of authentication. Following the guide for Certificate-Based Authentication on Palo Alto’s customer support portal - [https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g0‌‍‍000‌00ClJdCAK](https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClJdCAK) – the only issues that arose were input errors for trivial naming. For example, the certification name and the common name were different among the same certificate, so it was unclear which name to use for the “Distinguished Name (Subject)” during the Local and Peer Identification for the IKE gateway authentication. The Certification Name was used.

After the certificates were shared, the tunnel was established as expected and its functionalities worked just as well as pre-shared authentication.

**Conclusion**

Being similar to the previous lab in functionality and purpose, using the same configuration, setup, and knowing what problems we may encounter greatly aided in reducing the chances of errors. Isolating the configuration’s changes helped us narrow the focus of authentication methods and certificates, and it became much easier to recognize what change caused which issue, and thus this lab was able to move smoothly and quickly.