Site-To-Site VPN

*Pre-shared Keys*

Adv Cisco Cybersecurity – Lab 8

Jason Liu

Cybersecurity – Mr. Mason & Mr. Hansen

Period 5

*Lab 8: Palo Alto Site-To-Site with Preshared Keys*

**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’s PA-410 firewalls. Then, to monitor the session via an intervening switch to inspect the packet protocols for proof of encryption. The tunnel should be authenticated and secured with a common key entered manually.

**Background Information**

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.

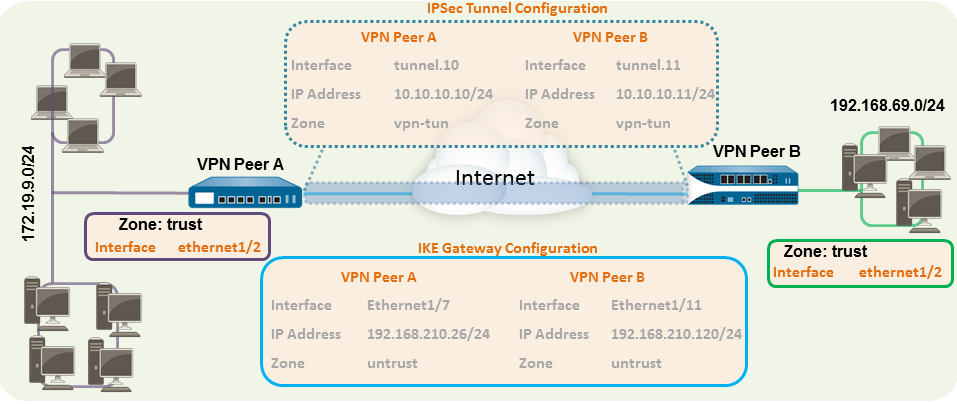
This image above describes the IKE messaging process, found on one of the research webpages: <https://www.techtarget.com/searchsecurity/definition/Internet-Key-Exchange>.

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**

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. Further details are described in the *Problems* section. The following topology was used:



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

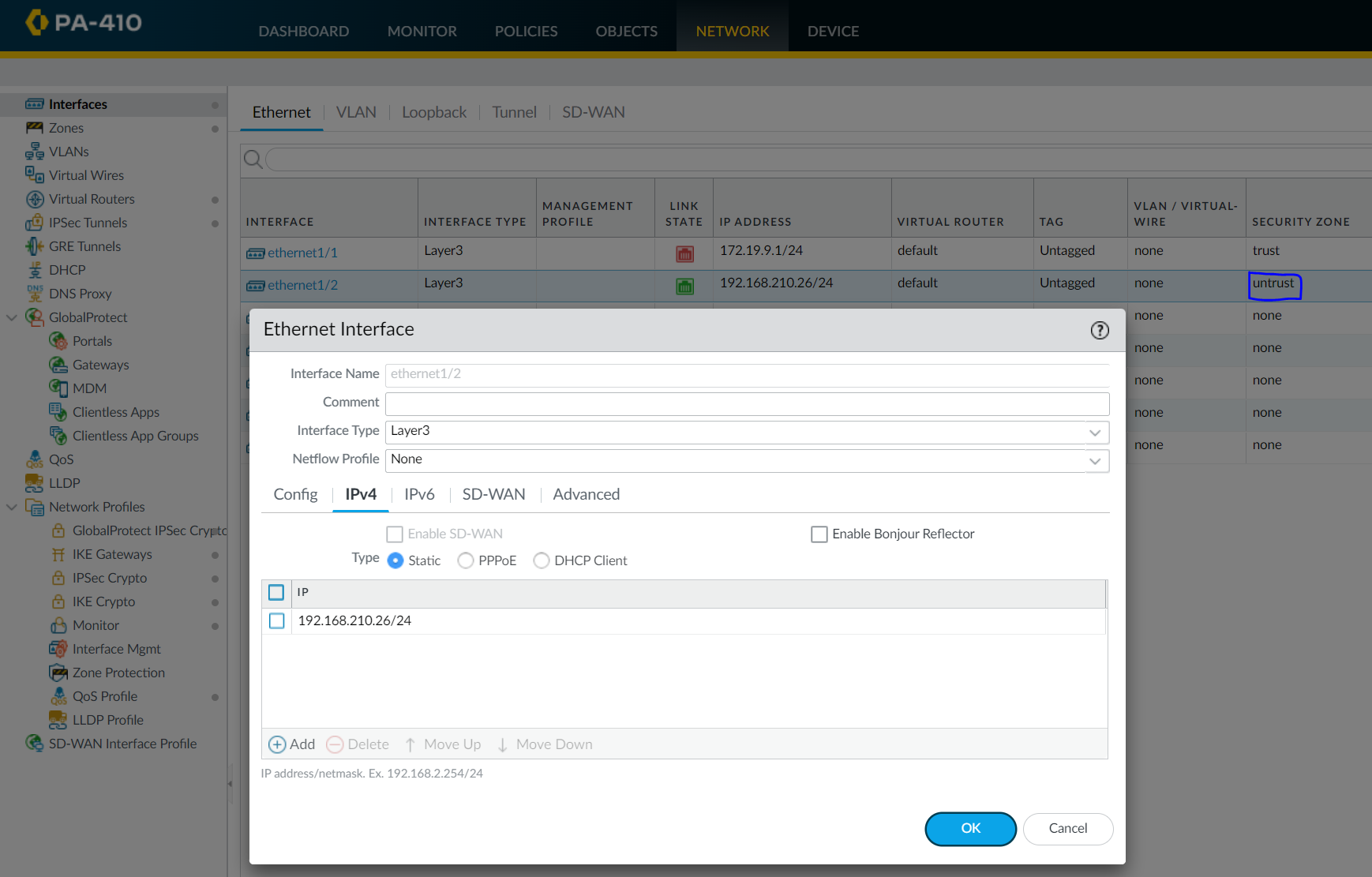
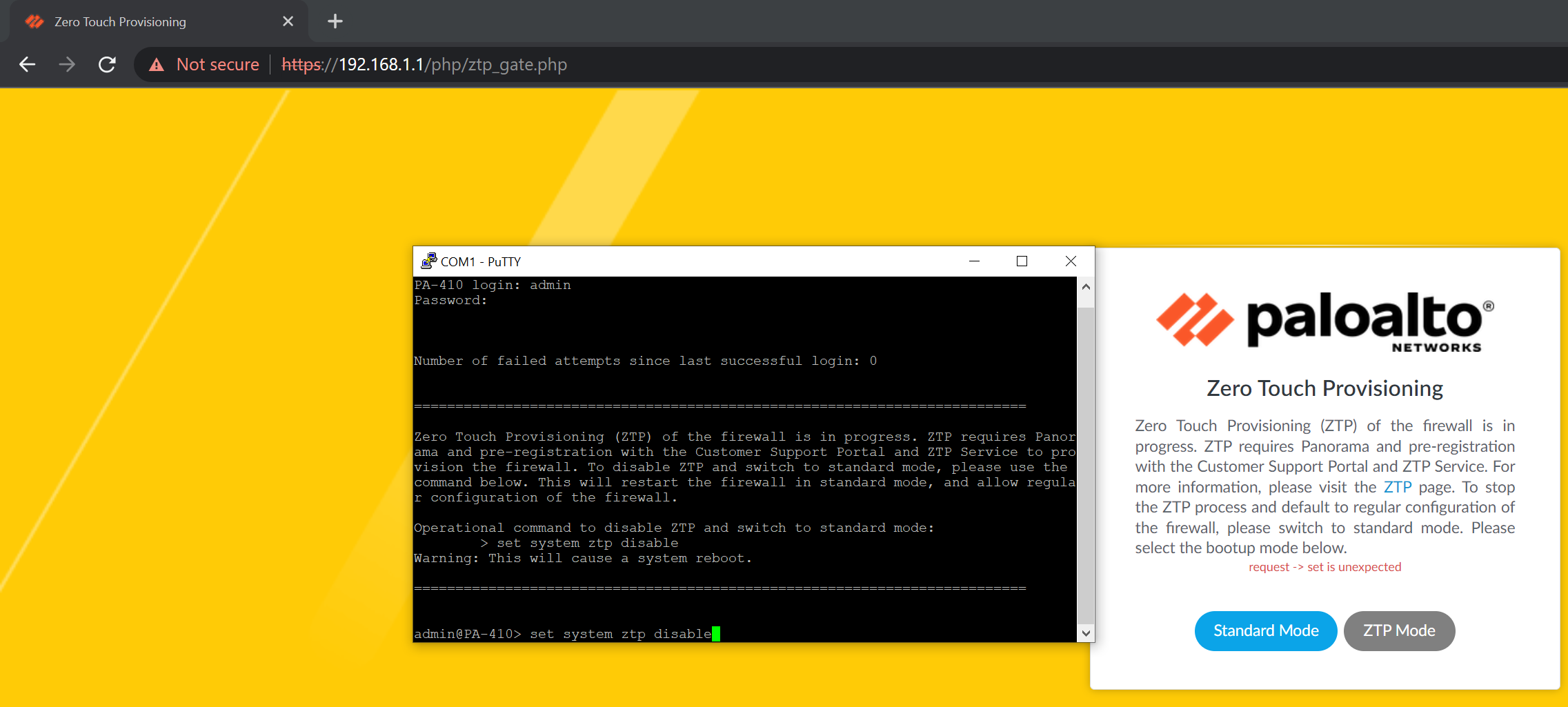
First, layer 3 ethernet interfaces have to be made on the firewall that correspond to their inside or outside security zones. These could also be named trust and untrust, respectively. One zone and one interface are unique, called the tunnel interface and the VPN tunnel zone, which acts as a virtual 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. A pre-shared key is manually inputted into the IKE gateway configuration. This pair of keys has to be identical to one another. The 8-character key *JasoEvan* was used in this lab.

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].

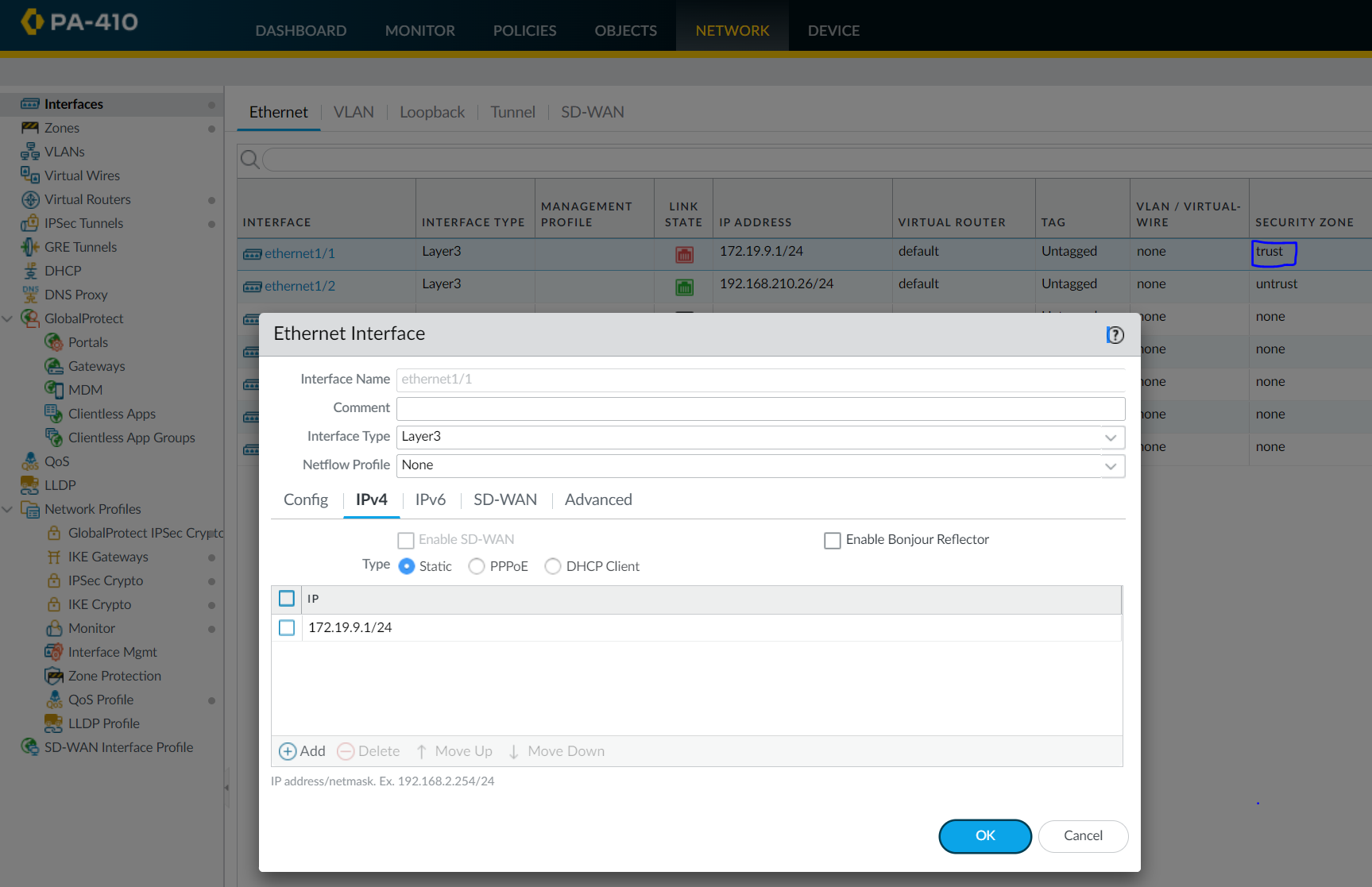
**Procedure**

The procedure is split into VPN Peer A and VPN Peer B,

****PEER A CONFIGURATIONS (See *Page 13* for PEER B)

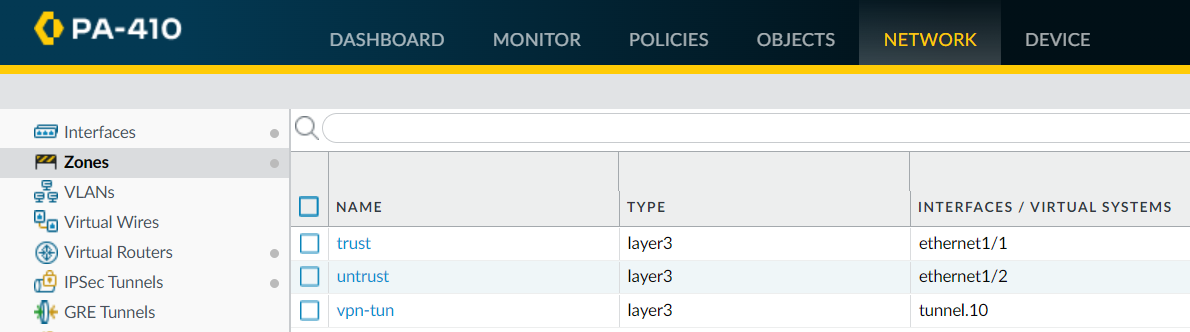
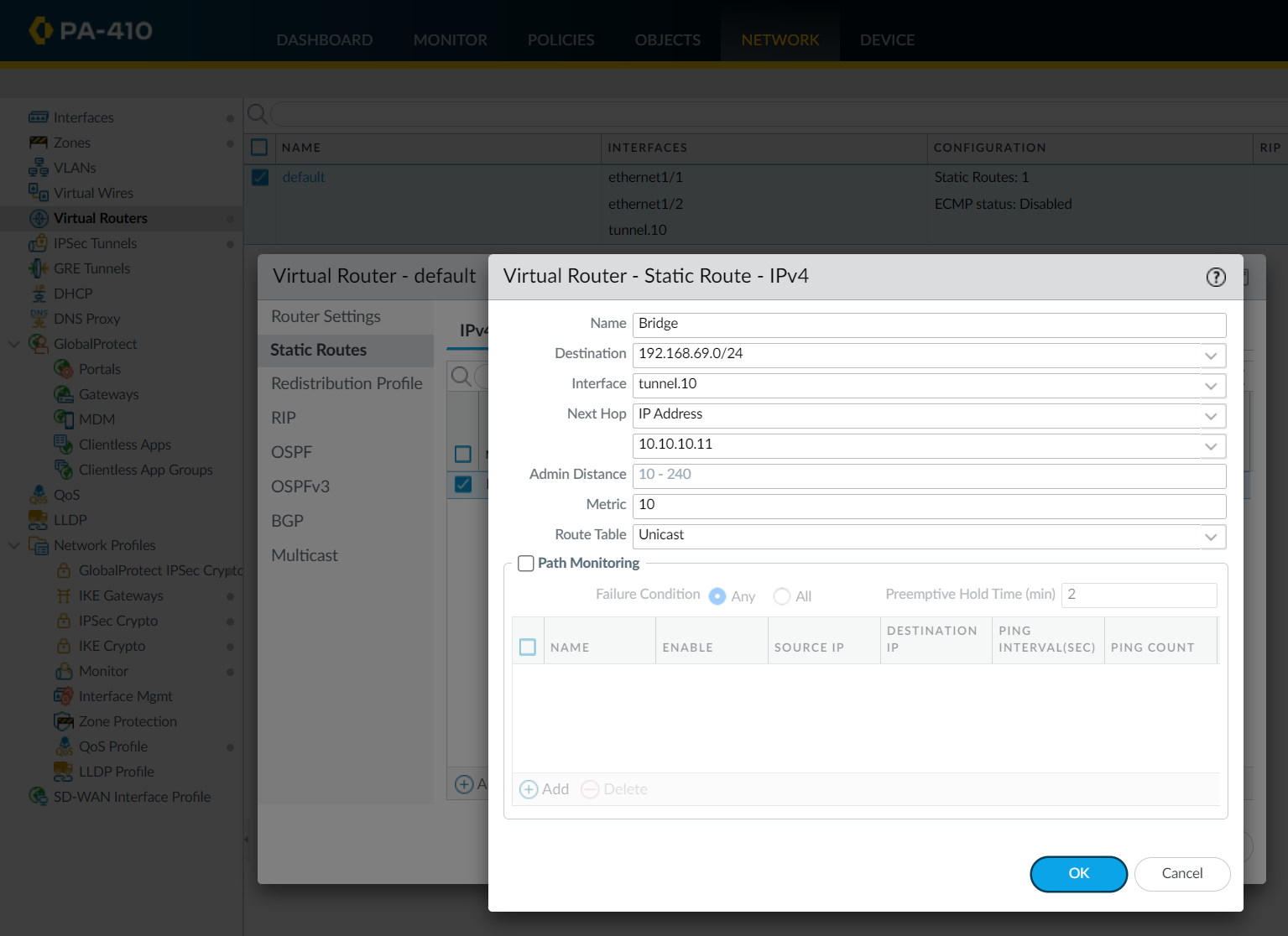
Create Outbound Ethernet interfaces and assign their interface addresses.

Disabling Zero Touch Provisioning to start the firewall in Standard Mode.

****

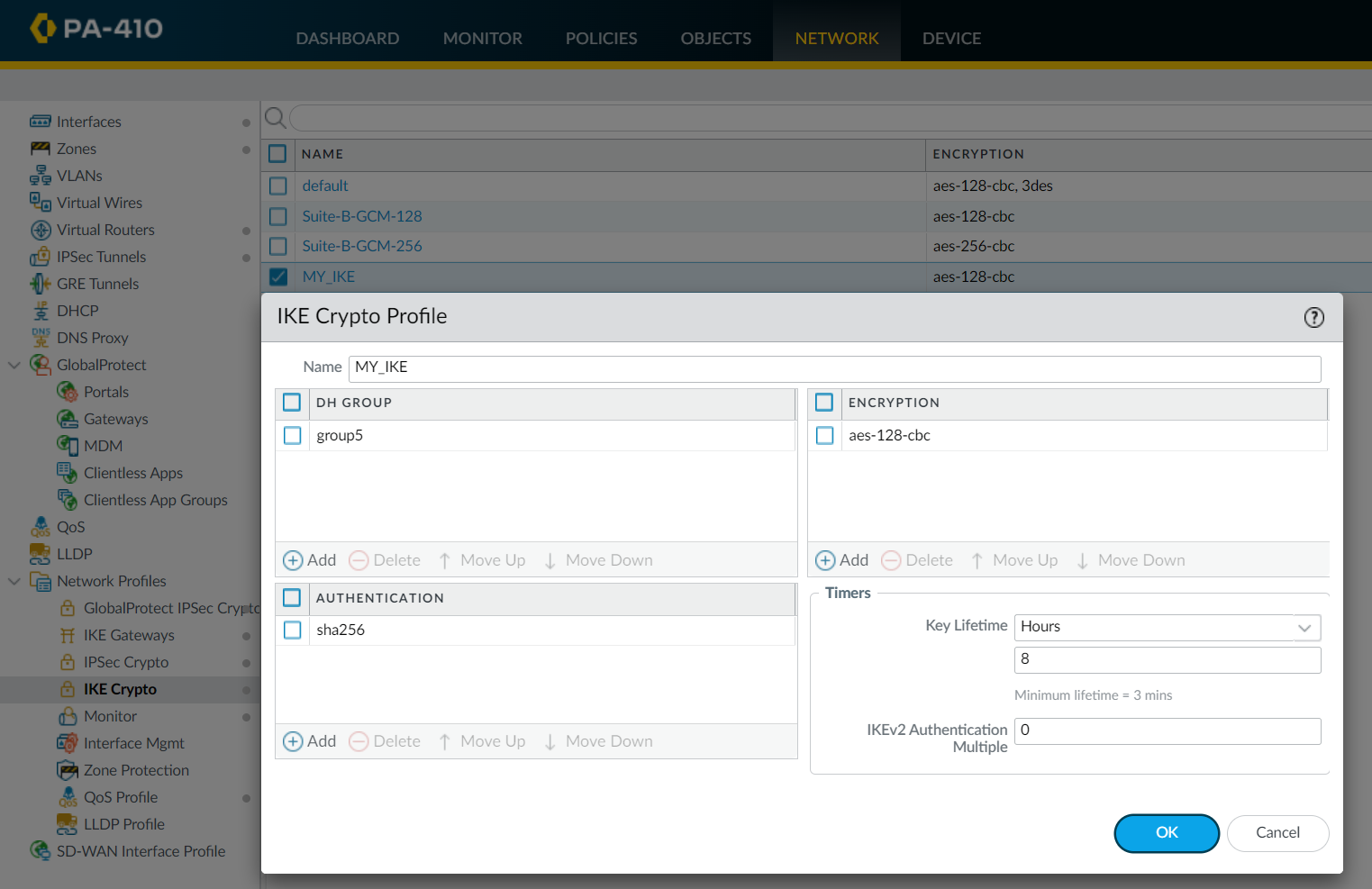
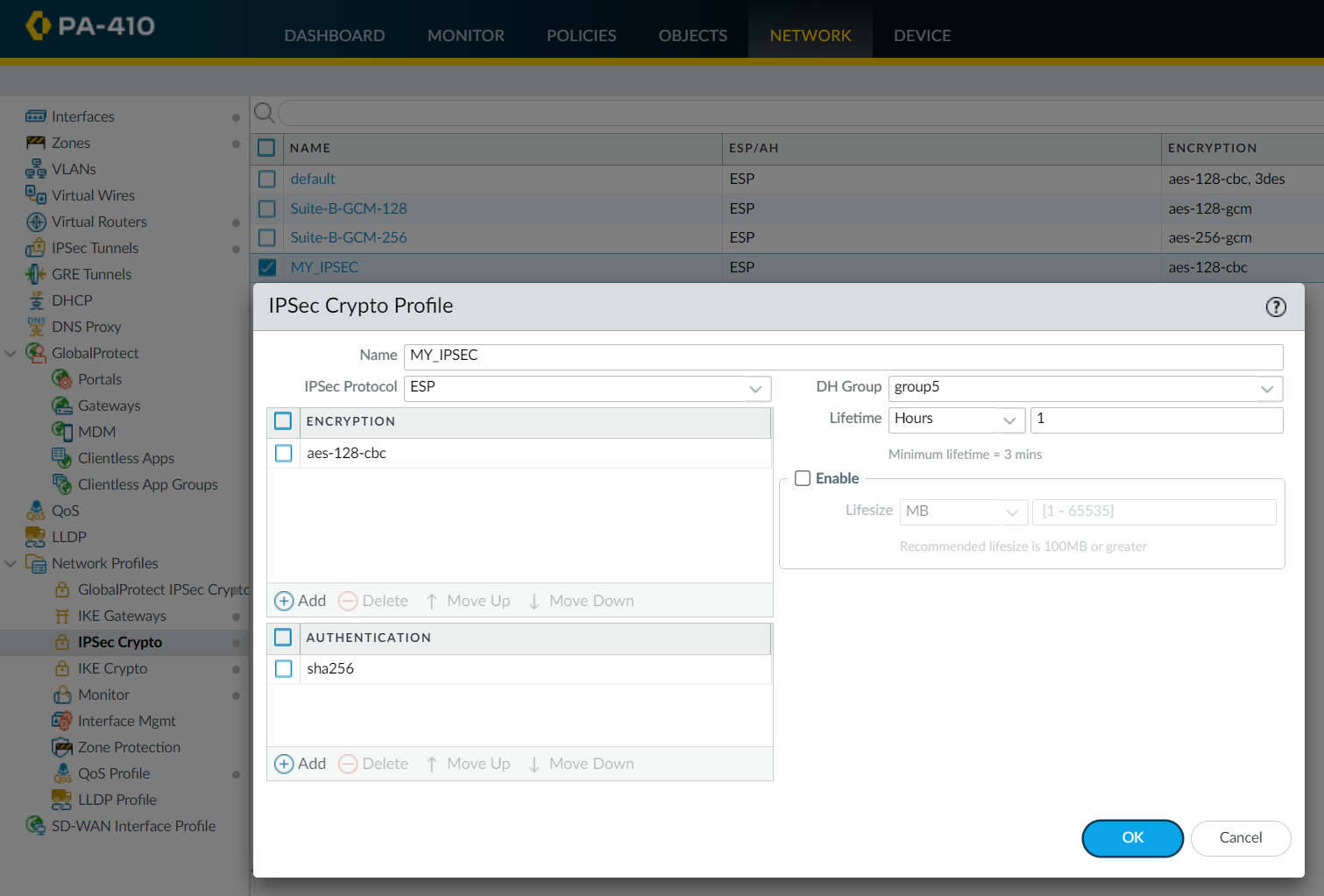
Configure the tunnel interface and assign it a tunnel IP. This address should be in the same subnet of the VPN peer.

Create the Inbound Ethernet Interfaces and assign them the local IP.

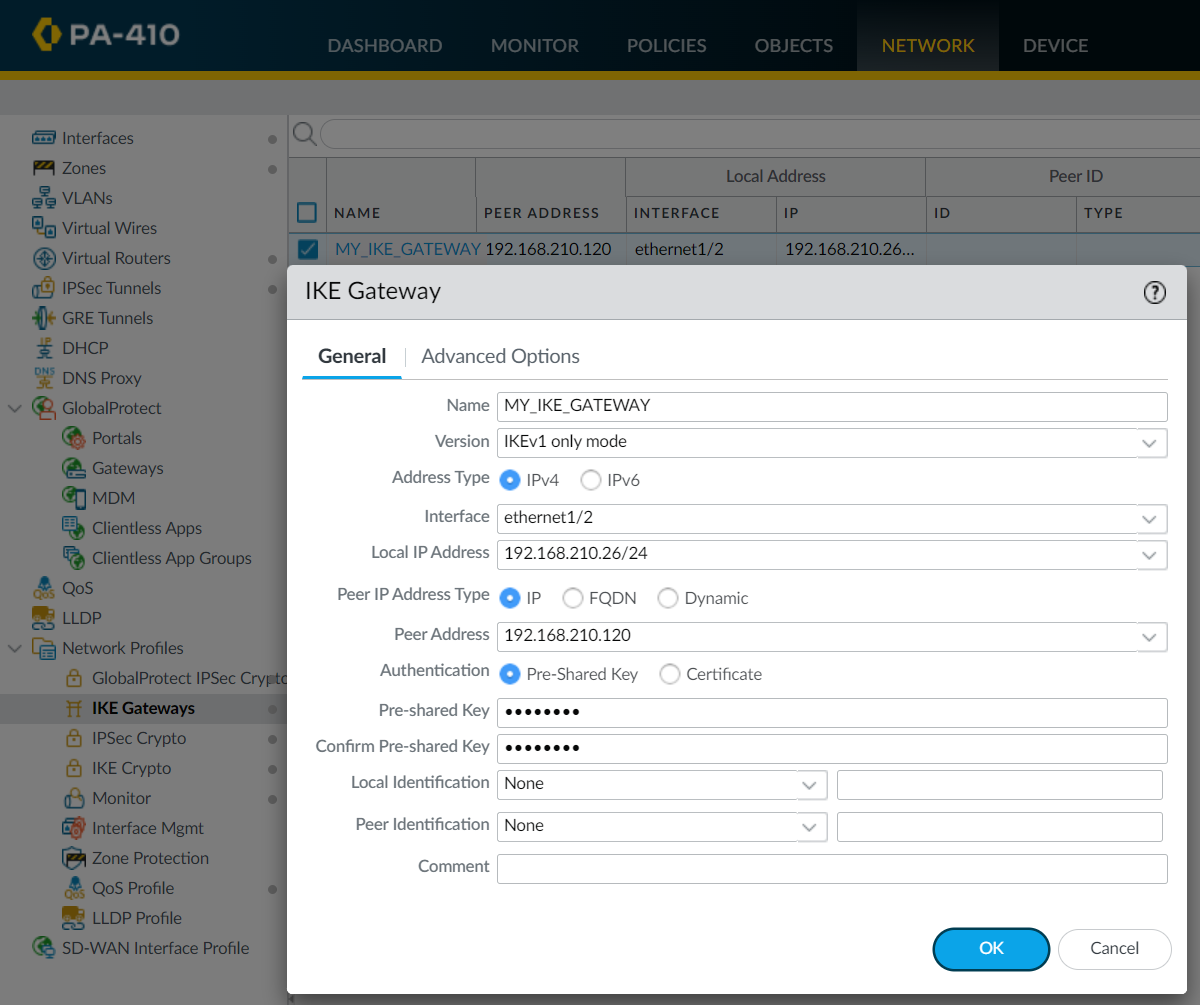
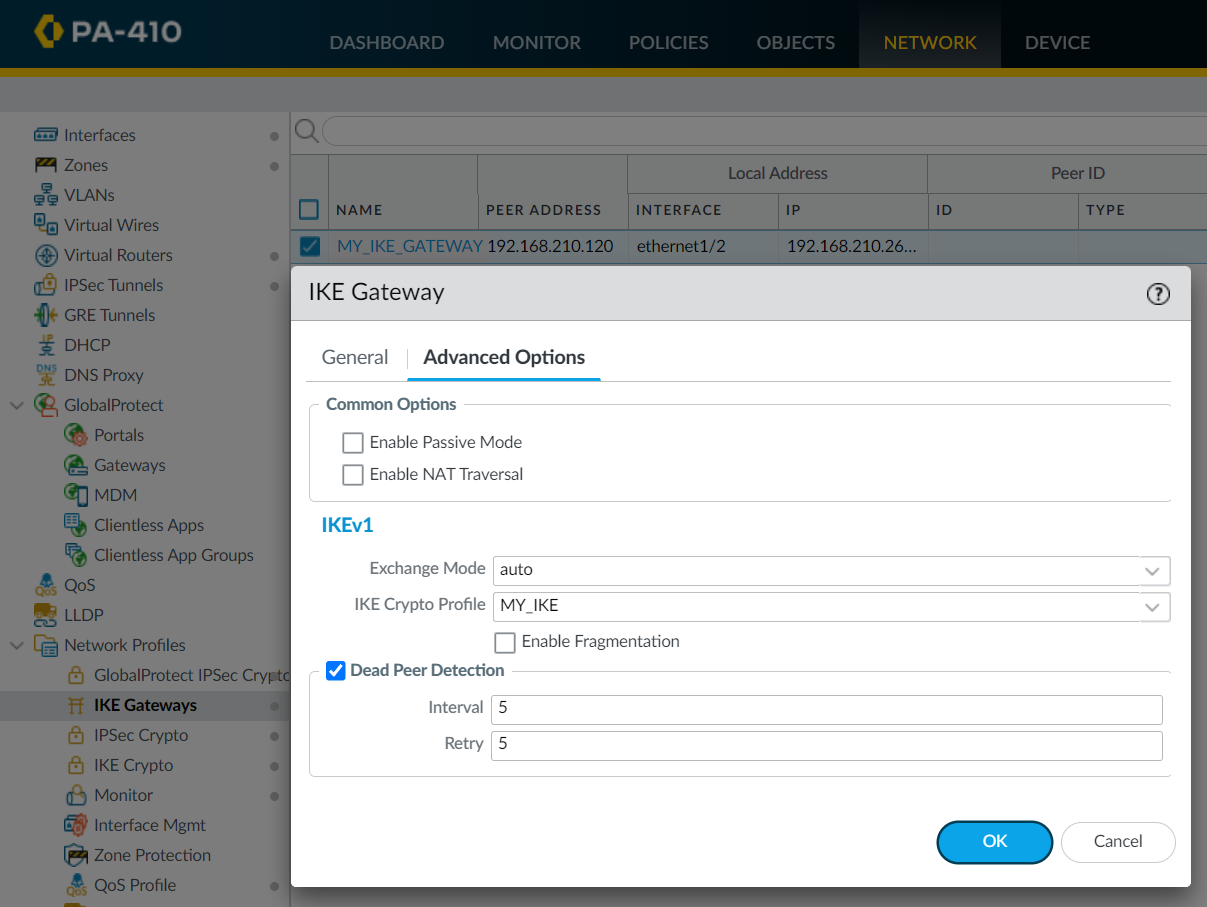
****

Route through the tunnel statically using the Next Hop IP of the end of the tunnel for addresses targeted towards the opposing network.

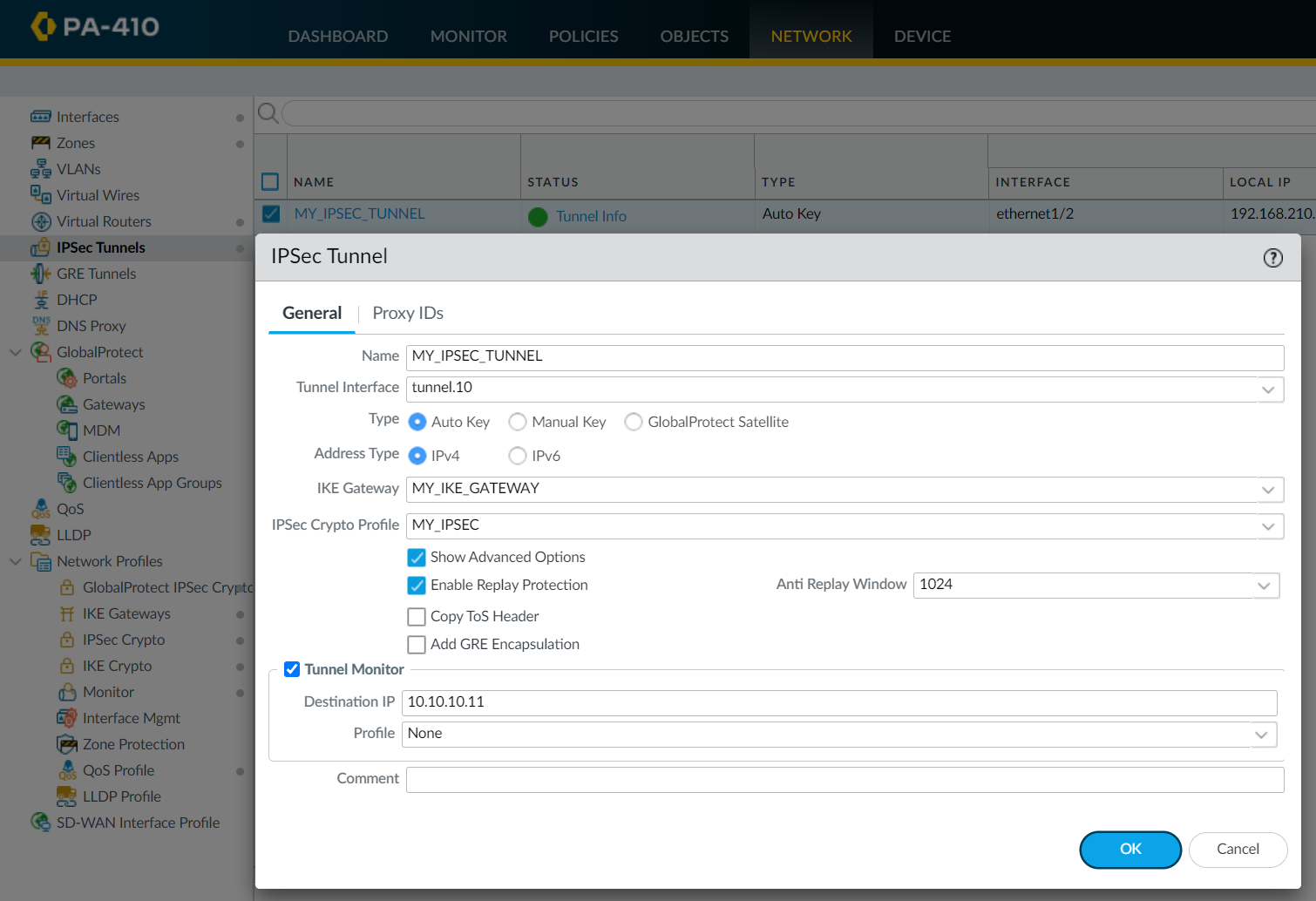
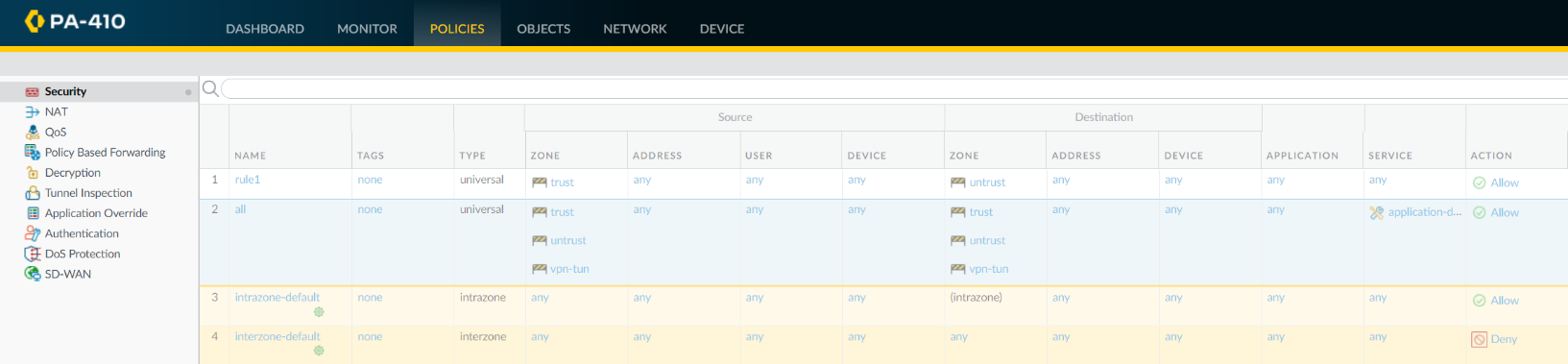
There should be 3 zones, each with its associated physical/virtual interface.

****

Create the IKE and IPSec crypto profiles with the same encryption and authentication.

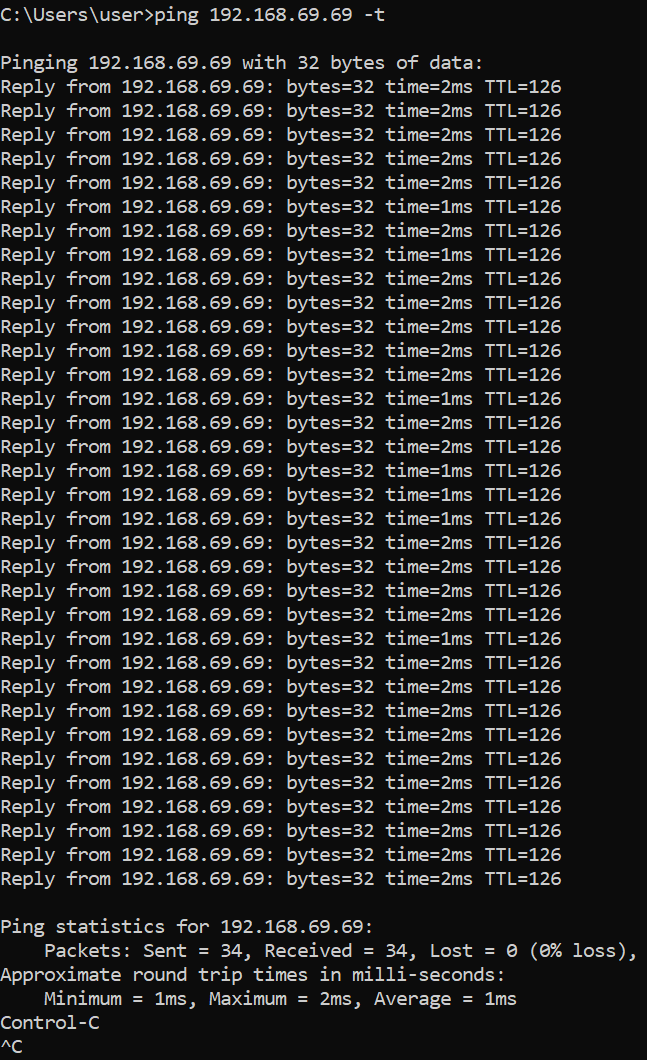
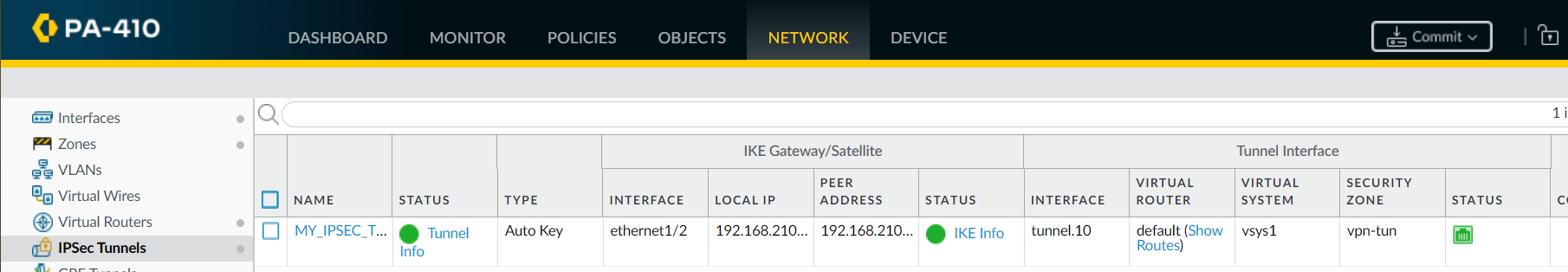
****

IKE Gateway configurations. This is where you input the pre-shared key for authentication. The 8-character key *JasoEvan* was used for this lab. Correctly target the peer address of their physical interface.

****

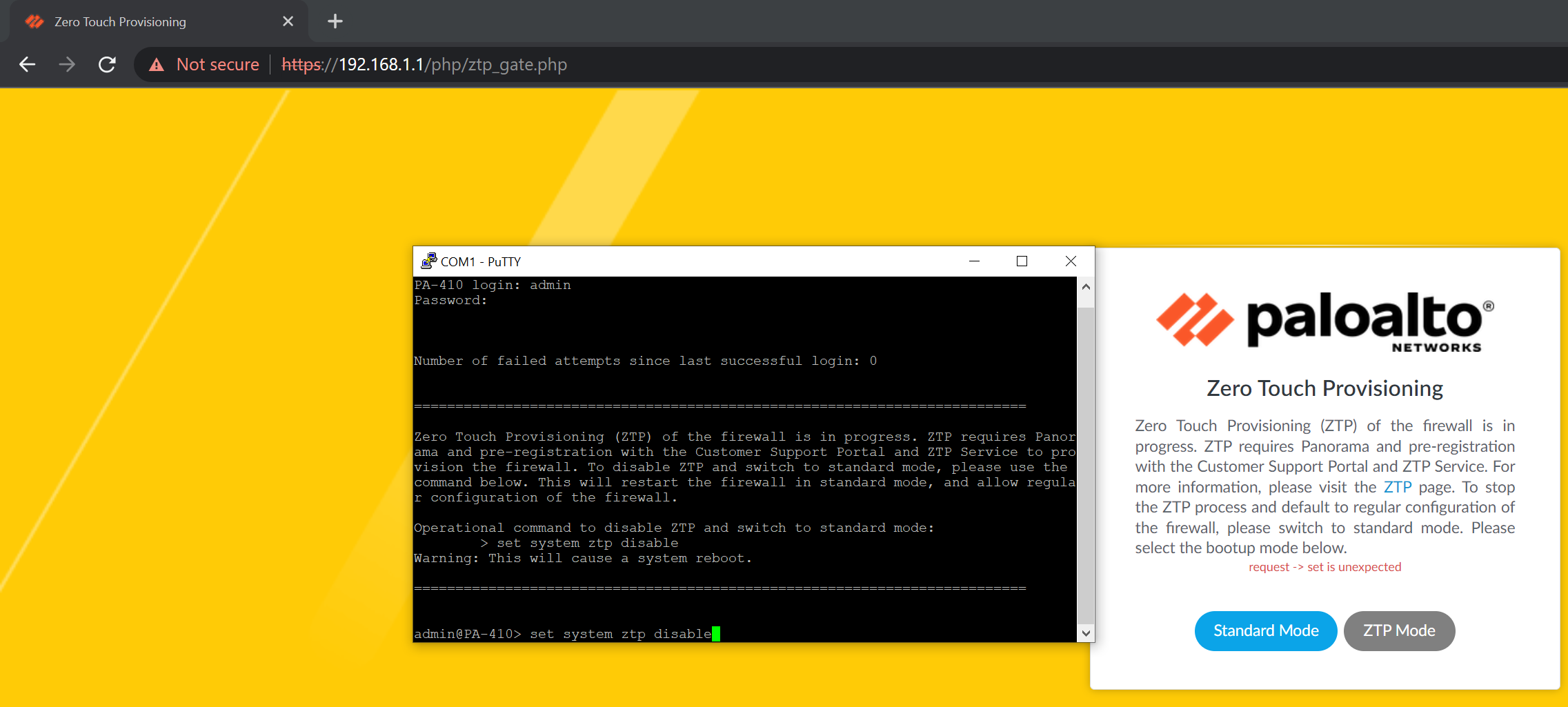
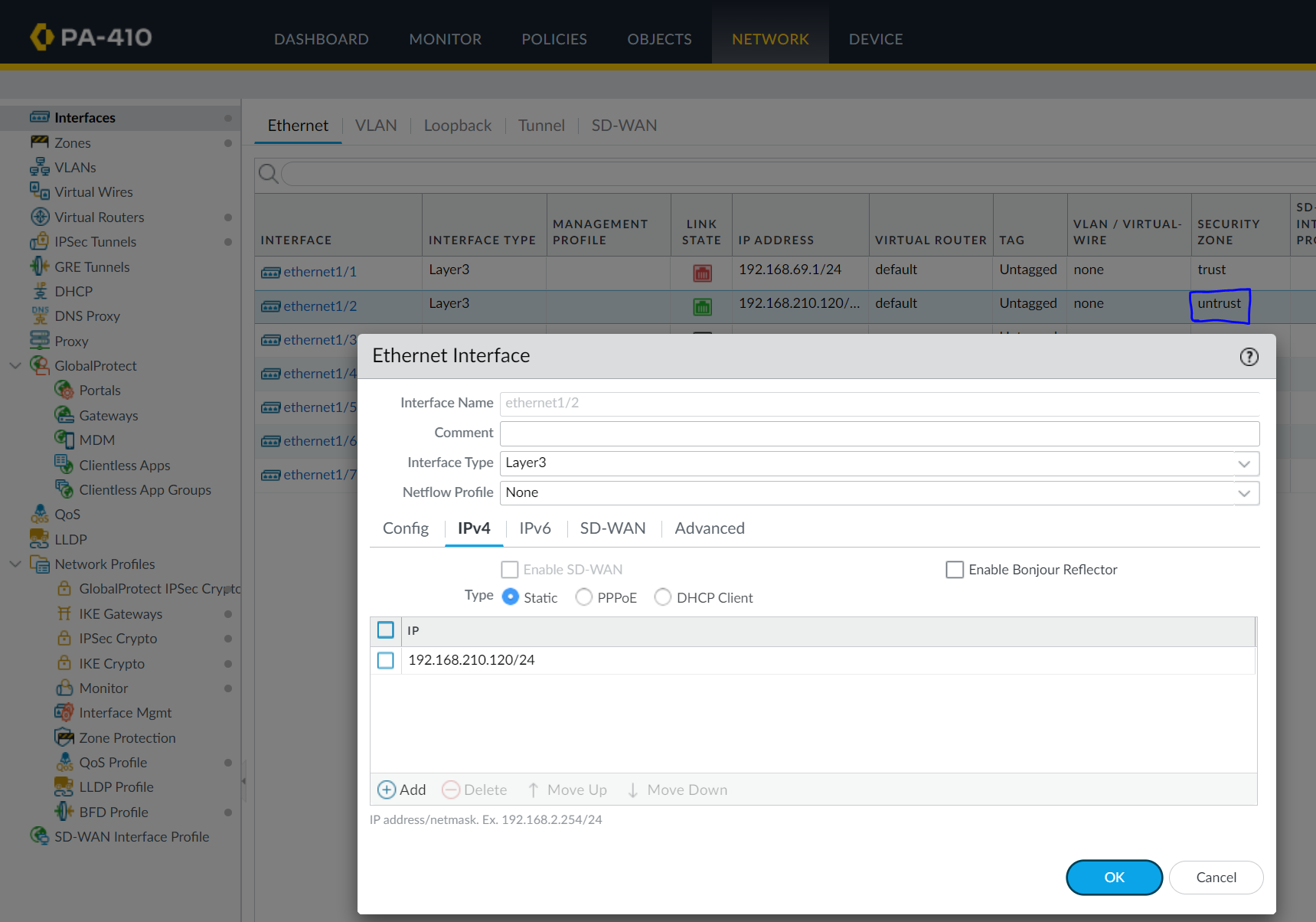
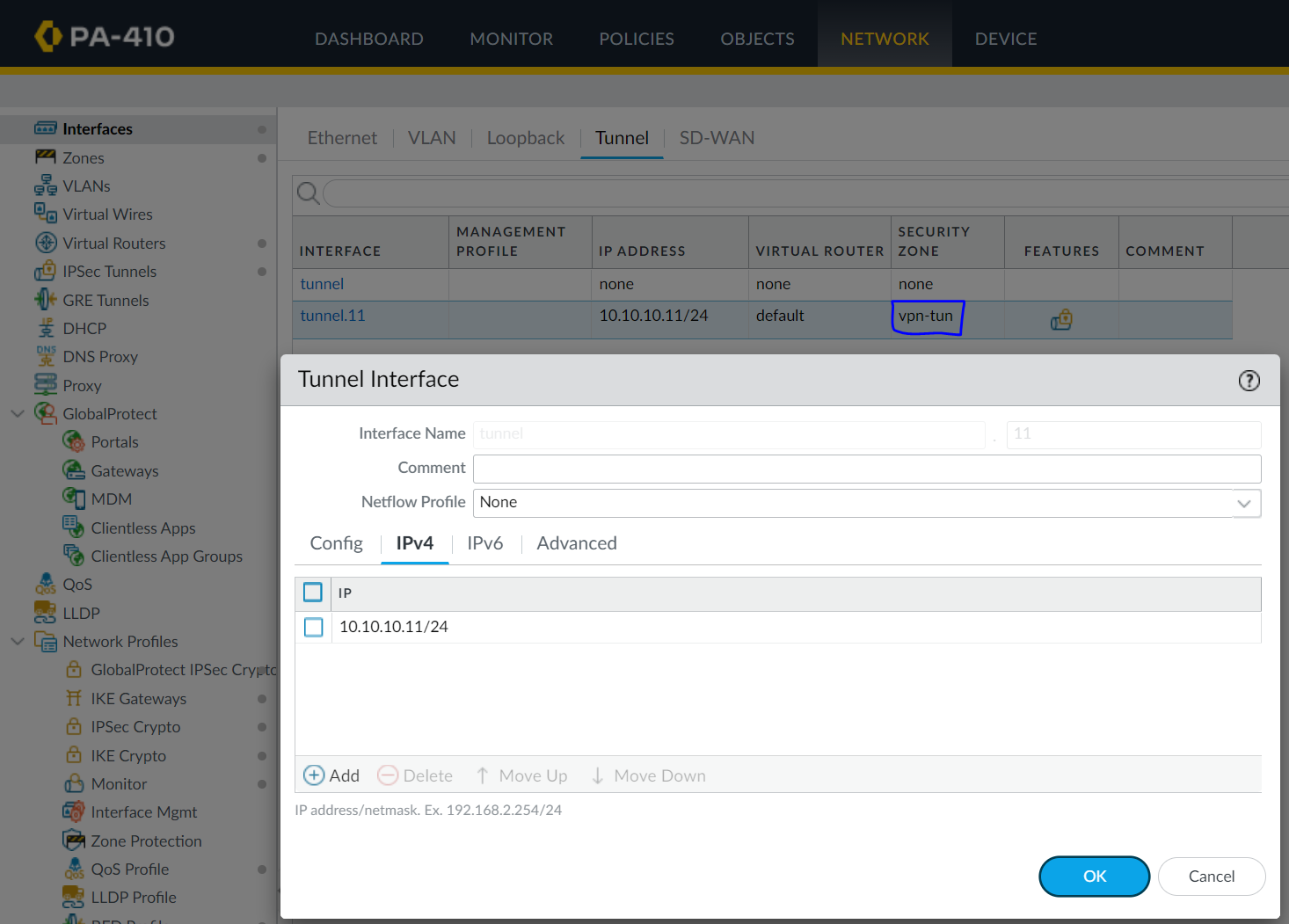
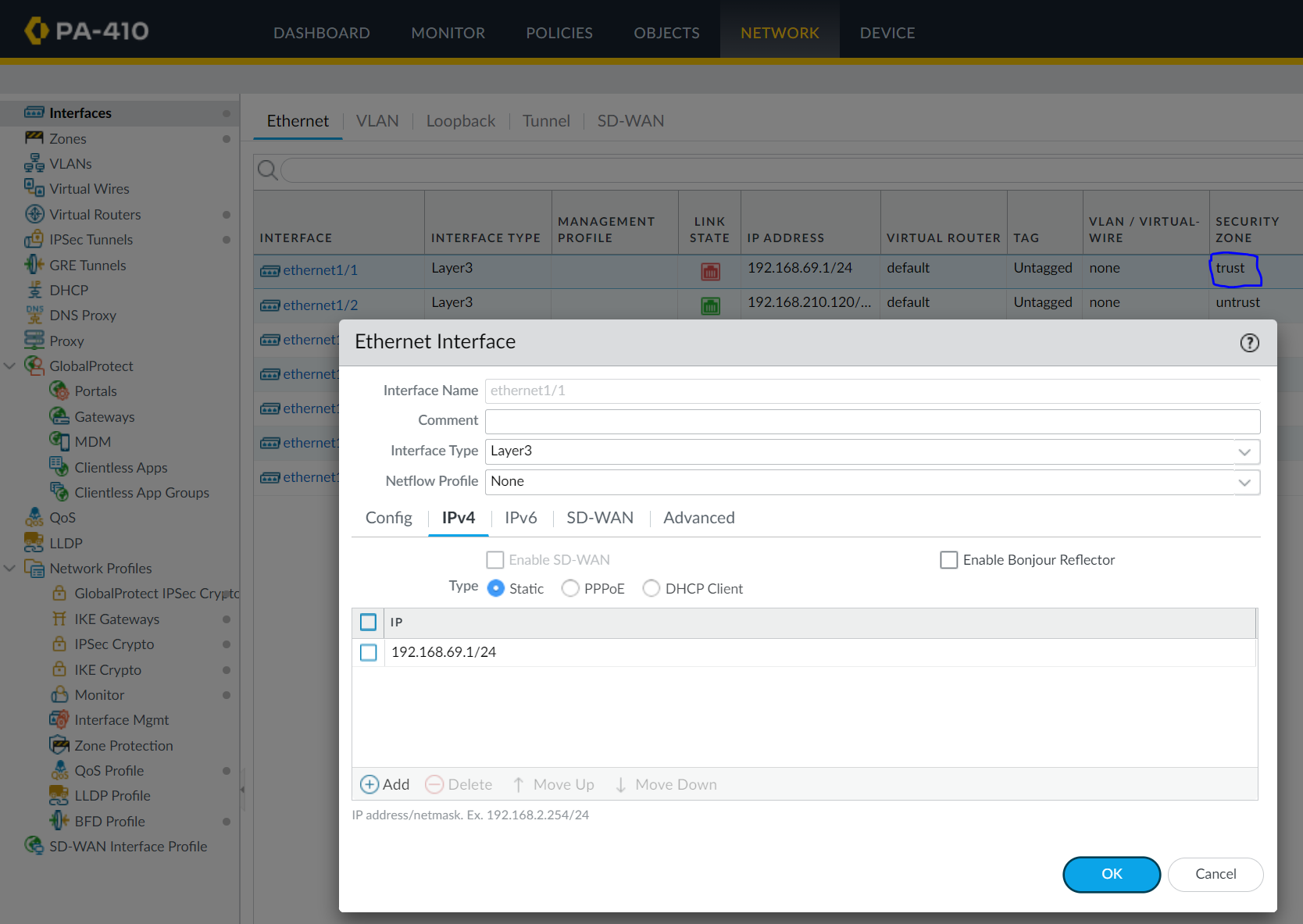
Create a Security Policy to allow traffic between all 3 zones.

Configure the IPSec tunnel to use all the previous gateways, interfaces, and addresses previously configured.

****

Continuous pings are sent from a PC in the inbound zone (local address) towards the opposing private address. The pings are successful.

The IPSec tunnel should be established and green. This means that the connection is formed and packets can be sent across encrypted.

PEER B CONFIGURATIONS

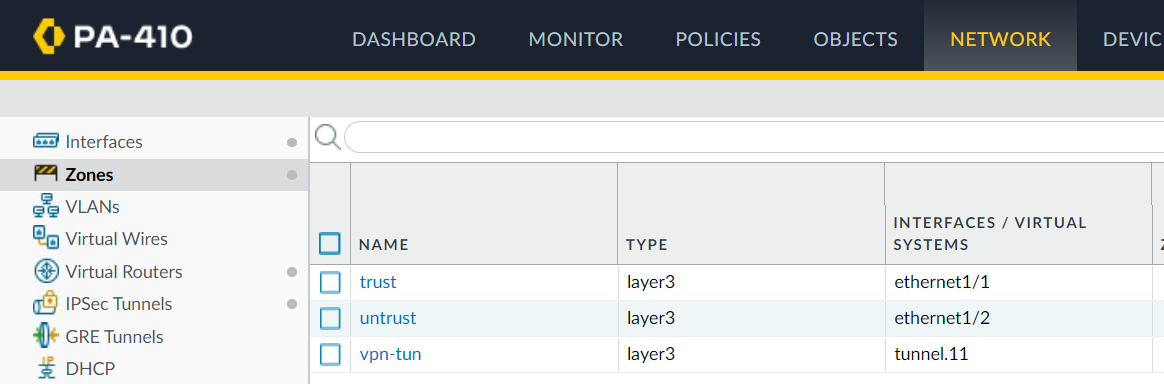
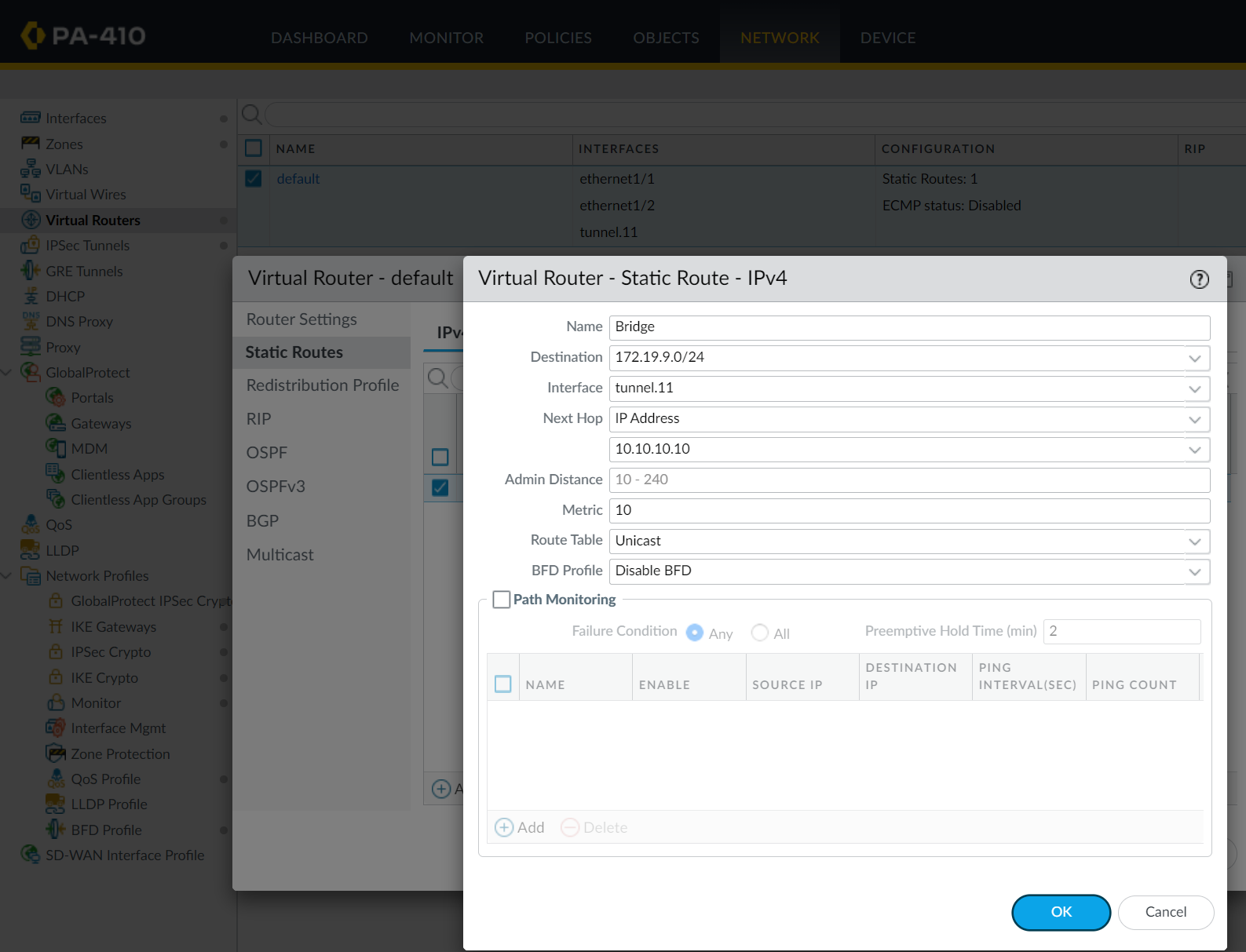
Configure Outbound interfaces and assign addresses that go out of the firewall.

Disabling ZTP and entering through standard mode.

Configure the tunnel interface. This is on the same subnet as the VPN peer A’s tunnel interface address.

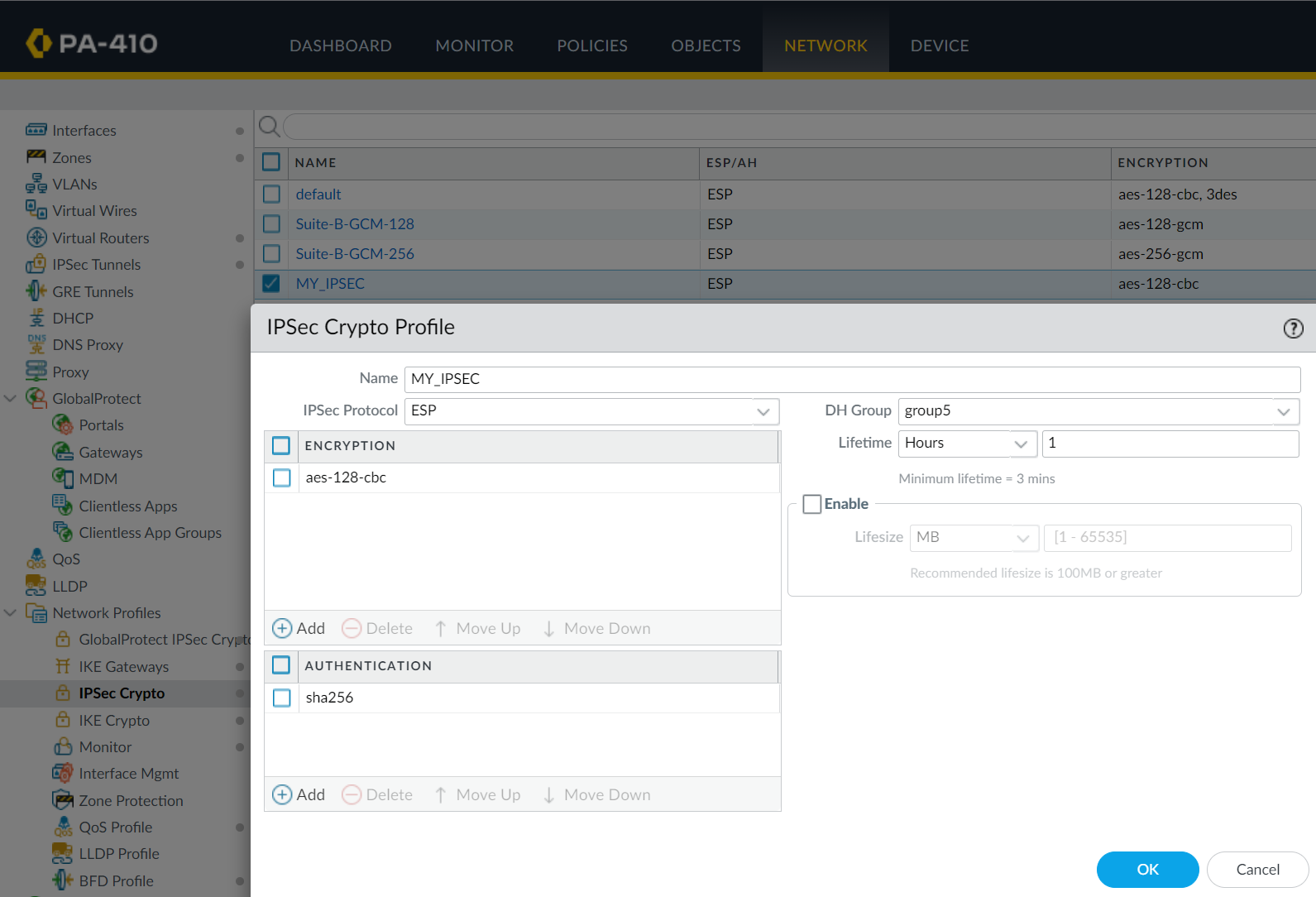
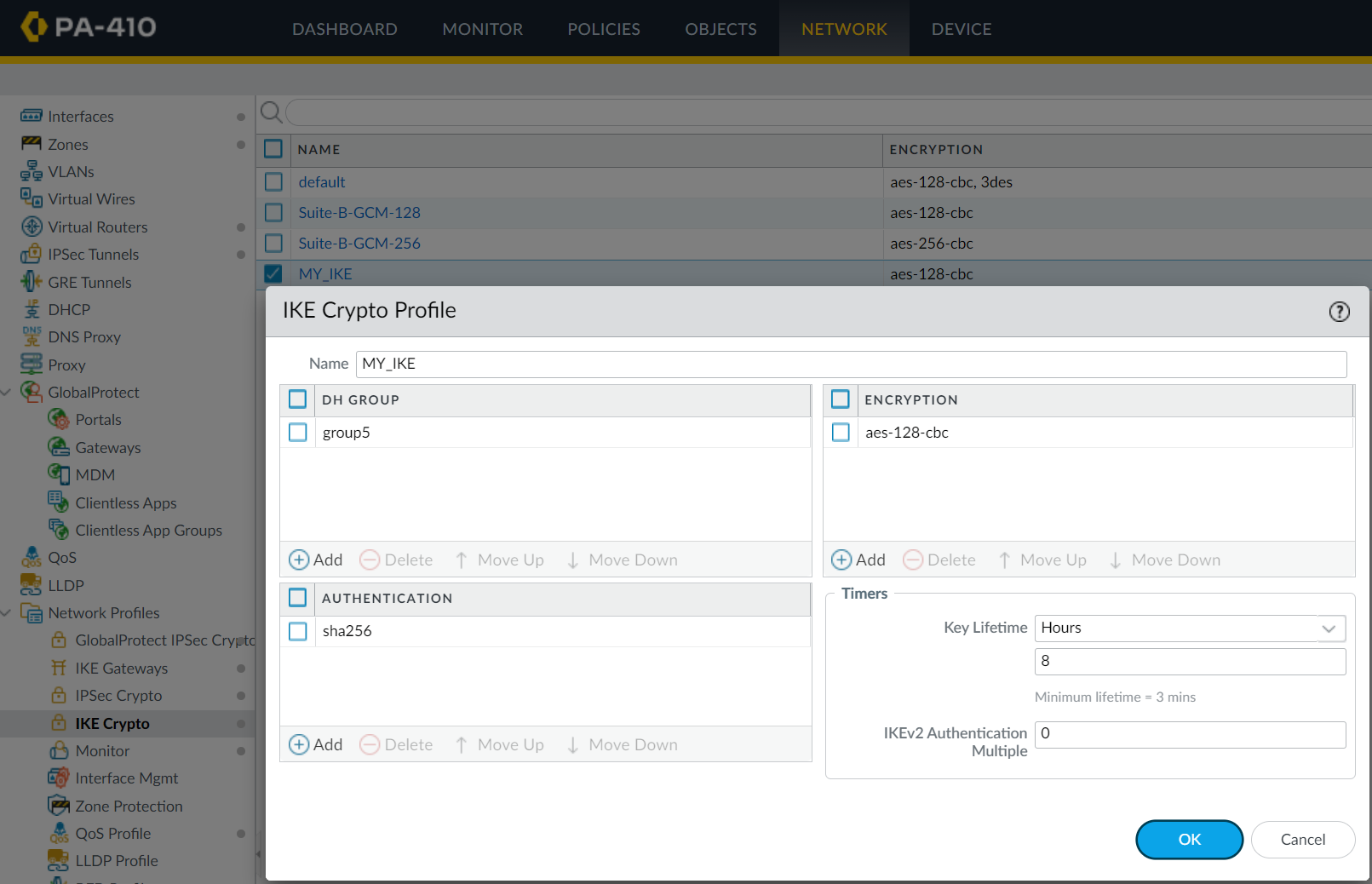
Configure Inbound interfaces. These are your local addresses and represent the private LAN.

The initial prompt. It will ask to change the password. Default credentials are *admin*.

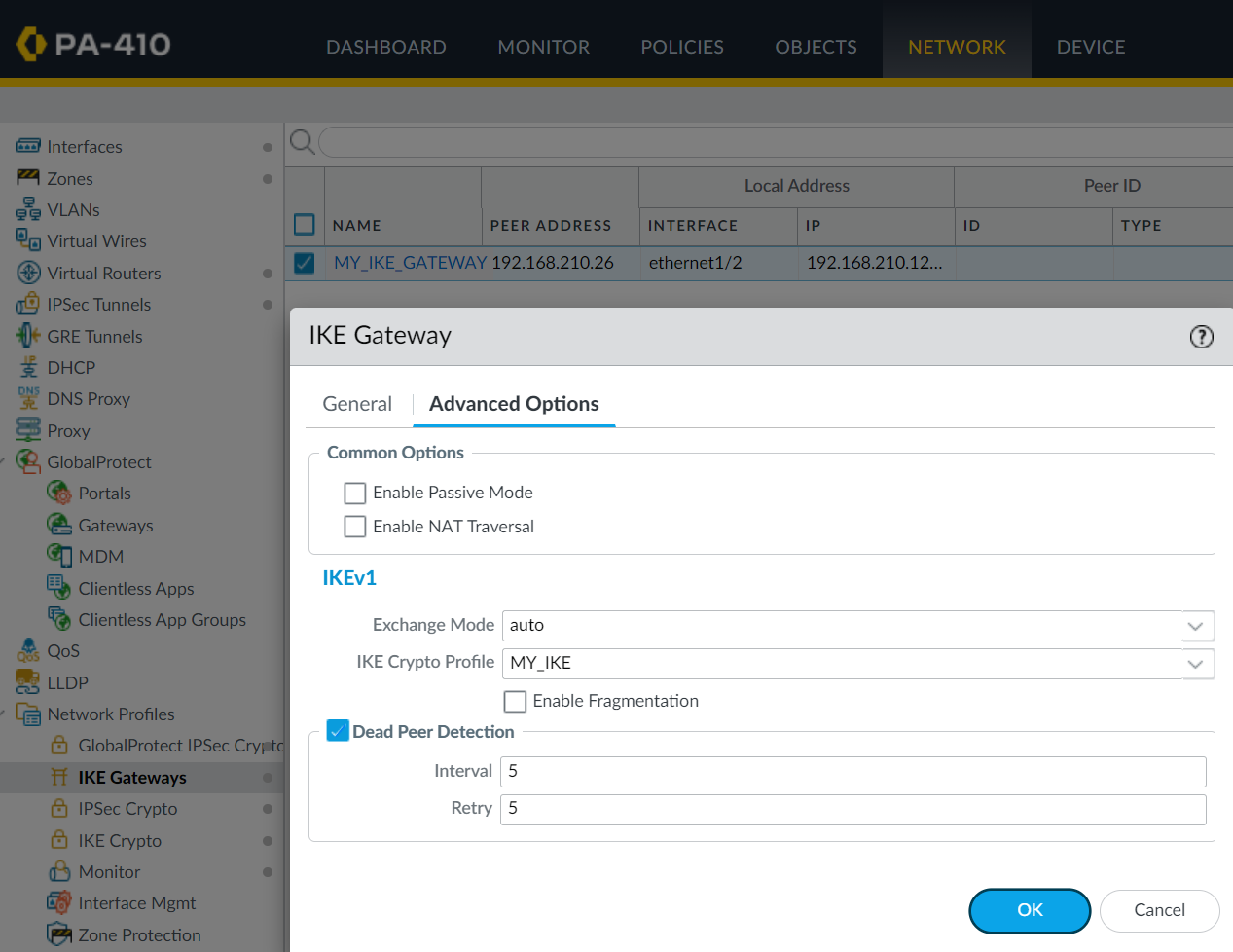
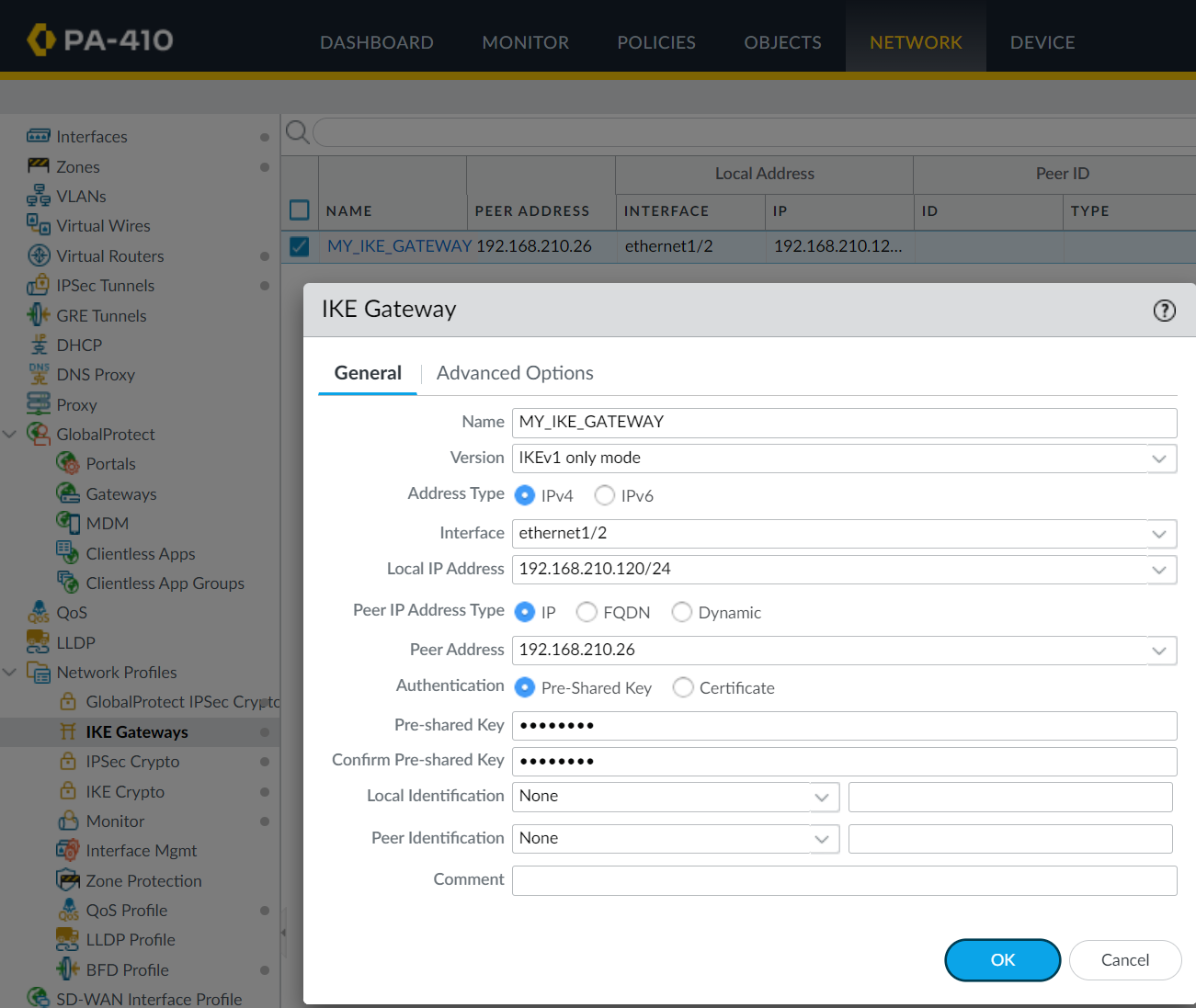


Statically route the tunnel interface to the VPN Peer tunnel IP for destinations that target the peer’s private subnet.

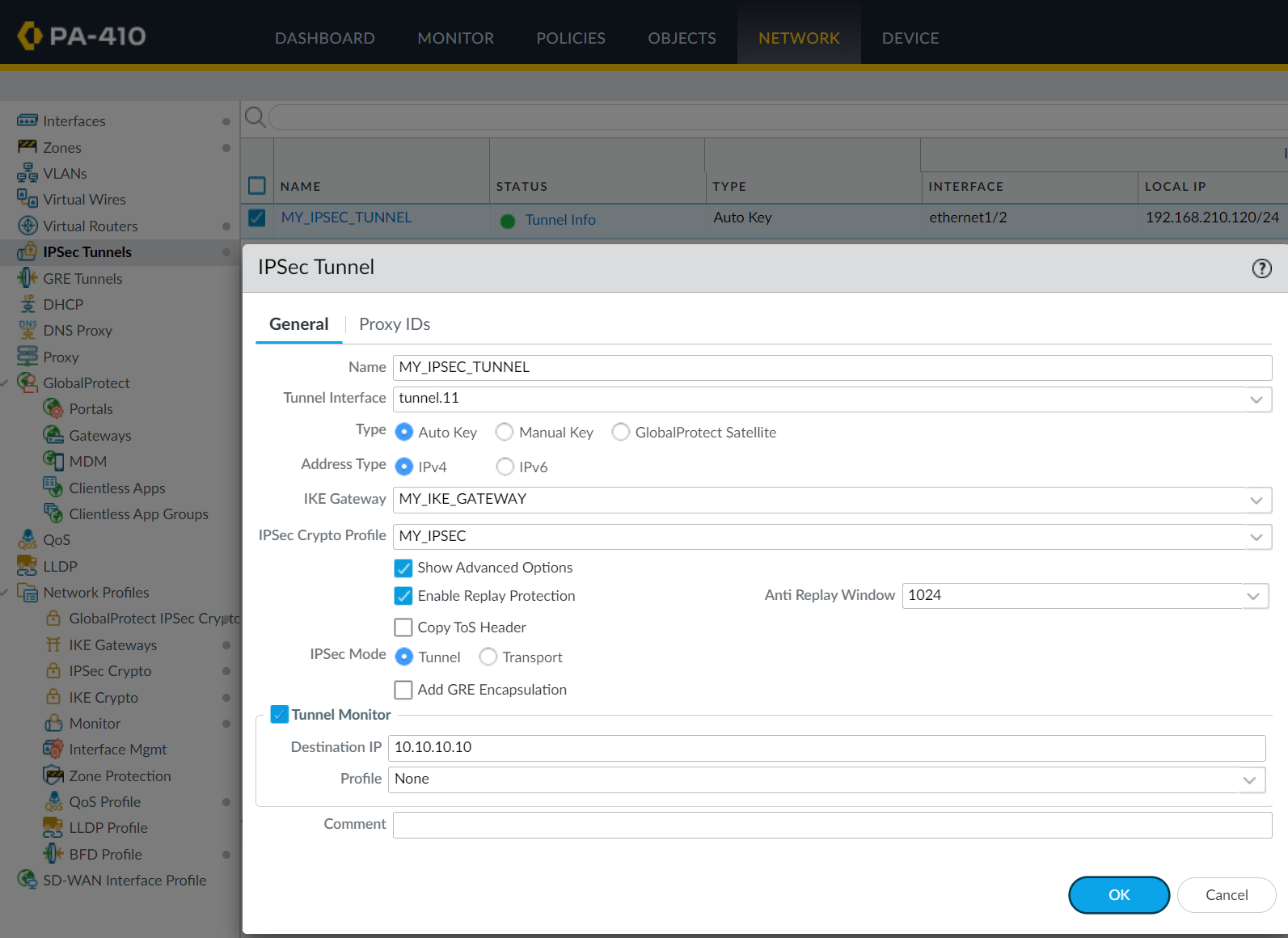
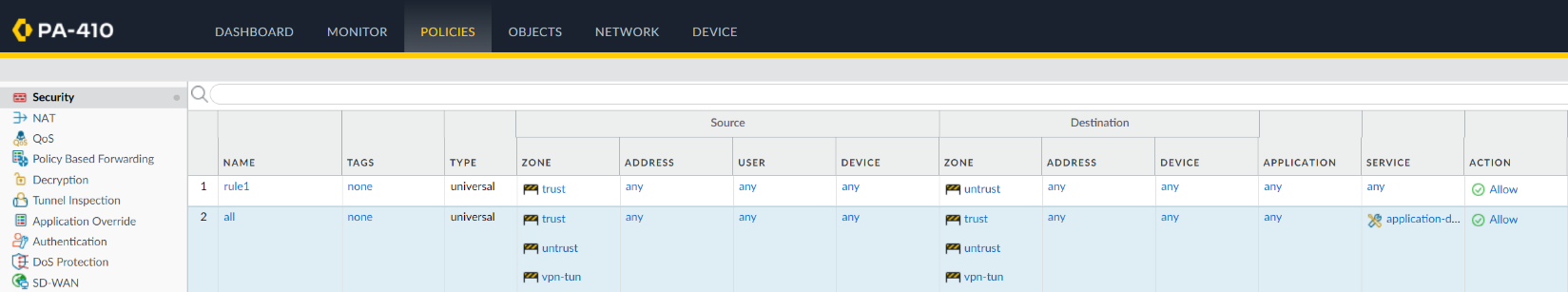
3 Security Zones created with their corresponding physical/virtual interfaces.



Create IKE and IPSec Crypto Profiles with the same encryption and authentication.

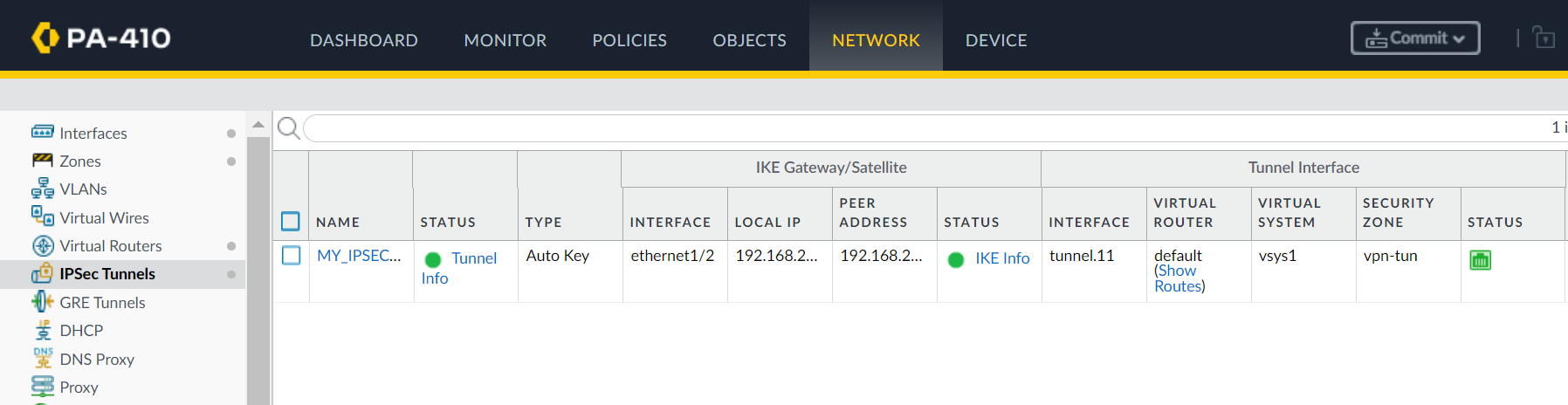


Create the IKE gateway with a peer address of the Peer’s physical interface IP. The pre-shared key *JasoEvan* has to be identical here compared to Peer A. Also note advanced options and other minor changes.

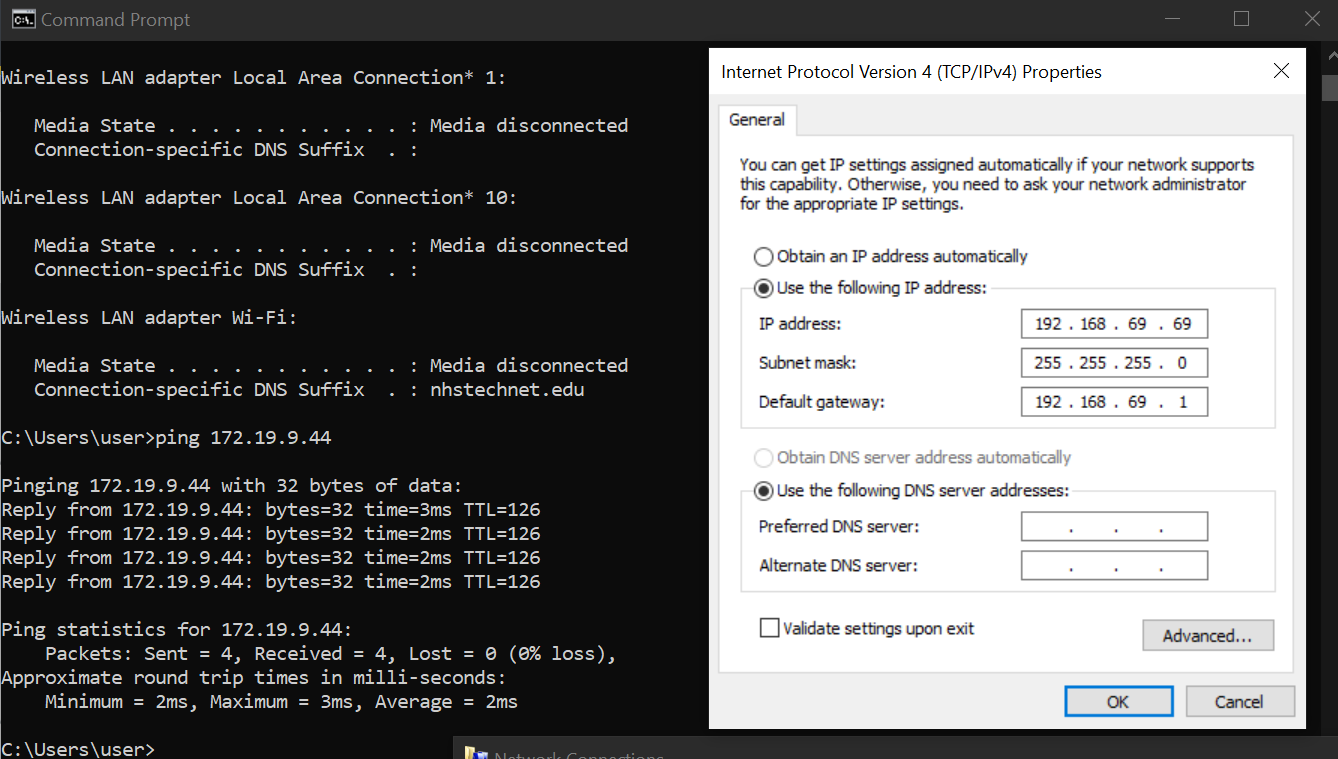


Configure the IPSec tunnel using the previous gateways and profiles.

Create Security Policy to allow traffic from all zones to communicate with each other.

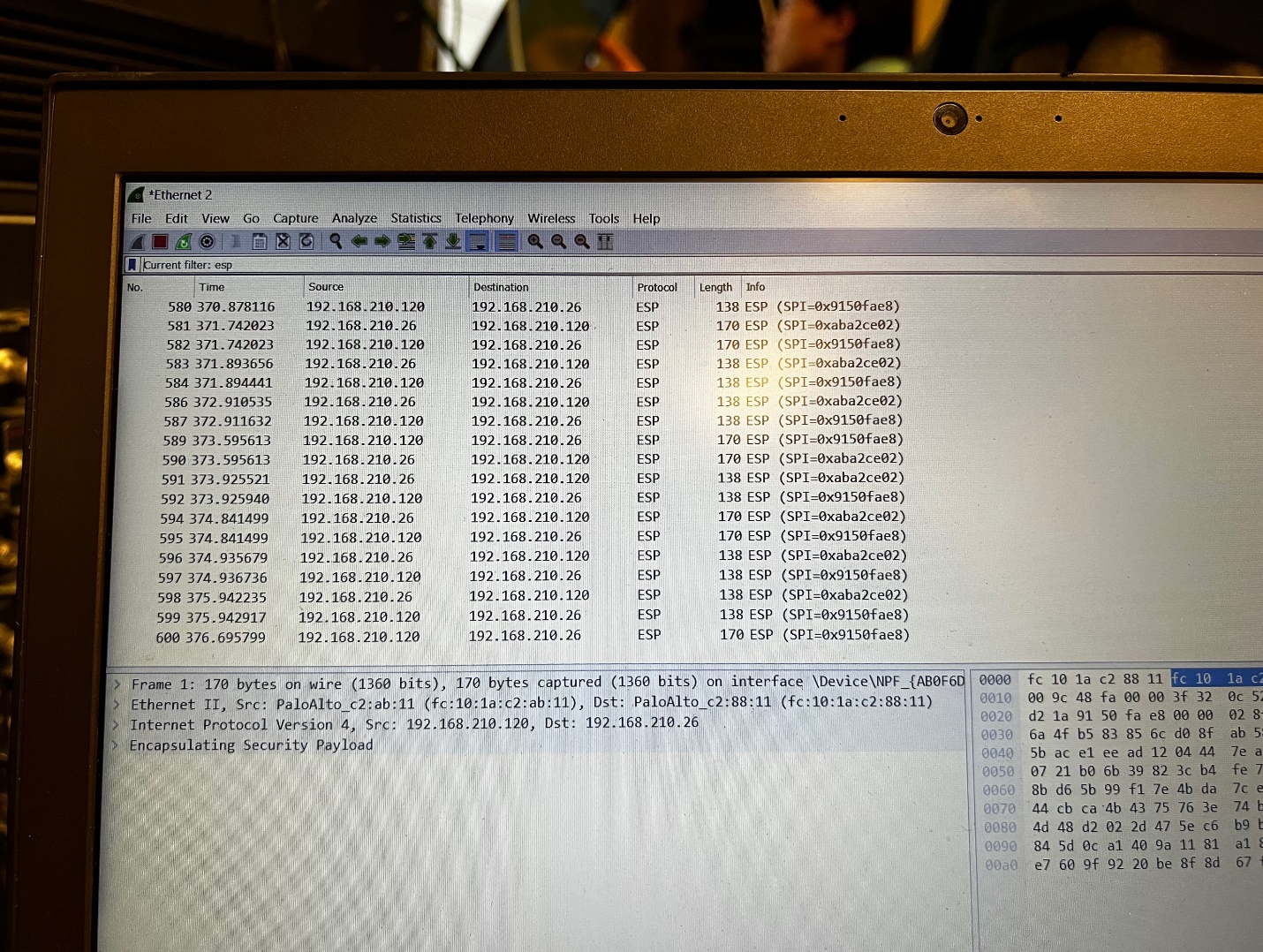


The tunnel should now be established and green. The connection is formed and packets are encrypted over this tunnel.



Pings across the clients on the private LANs of opposing ends of the tunnel can properly ping.

PROOF OF ENCRYPTION

****

Under a monitored session in WireShark, the packets are read as ESP protocol, meaning that they are encrypted. Monitoring Session means that a copy of the packet that comes from the “source” port is not only sent to its original destination but also to the “destination” port of the monitored session.

This means that a copy of the targeted pings are sent here to this external laptop and are indeed encrypted.

**Problems**

This lab took a long time. A document including all attempts, solutions, and sources have been recorded alongside the progress of this lab. An overview will be described here.

First, TLS, or transport layer security, has been an issue since the previous labs dealing with entering the management interface. Since the TLS version was changed in attempt to access the Cisco ASDM for the Cisco firewalls, trying to access the webGUI led to a 404 error page. This was temporarily solved by using a different drive while I would reimage my own SSD drive. However, it turned out the secure protocol HTTPS on Google Chrome allowed the changed version of TLS and for us to access the webGUI of the Palo Alto 410.

This next issue is a reoccurring hiccup that is quickly solved but often forgotten. The factory-reset PA-410 has many default configurations on it, such as a couple ethernet interfaces, a default virtual router, but also a virtual wire. After completing and changing the prior configurations for the new lab, the virtual wire is unnecessary. The hiccup is that the firewall cannot commit changes with virtual wires with missing connections, and so, the unneeded virtual wire can be quickly deleted to solve this issue.

The following problems occurred during the first IP scheme, following a configuration suggested by this video - <https://www.youtube.com/watch?v=jGh7ZPyqMHk>. This addressing scheme worked for a few other lab groups, and so we attempted this schematic and tutorial as well, with one firewall being the PA-220, while the other being the newer PA-410. However, we faced many issues.

During the first week of trials, our lab rack had access to only one computer, rather than the usual two. This meant that we could only configure one firewall’s webGUI at a time (as no SOHO was setup), making it difficult to simultaneously compare and check configurations for congruency. One such fault is that we had to change the authentication from md5 to sha256, which are authentication protocols with different intensities of security. We also had to change the encryption method from DES to AES, with at least 256, meaning 256 bits. Going back and making changes it extremely dangerous, especially on only one computer, as it means that you have to constantly make sure both are correct, committed and changed, and all other changes are also paired. We fear that we have forgotten to make a couple changes to one and not the other over the course of the process, which is why the tunnel was not established when we finished the configuration steps.

The most common issue is to make sure the pre-shared keys are identical, considering capitalization sensitive. It was also distracting that, in Palo Alto’s attempt to increase security, always show that the password is 8 characters, regardless of the inputted password’s true length. This made us believe that are password was incorrectly typed. After many minor shifts and changes, the tunnel was never established.

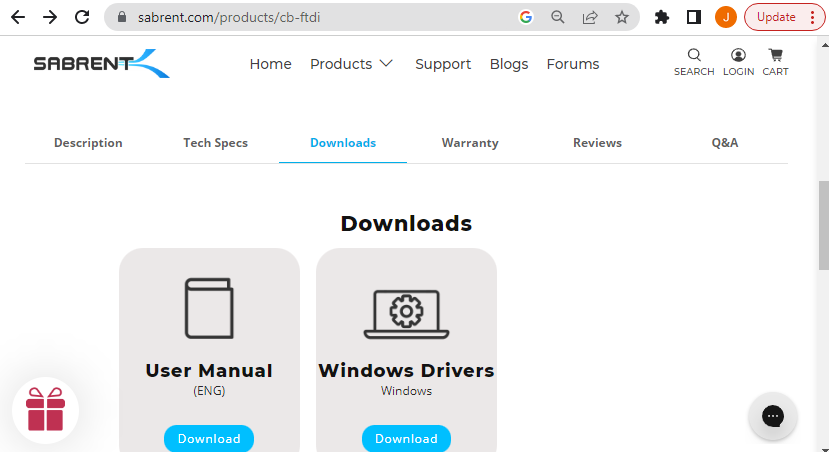
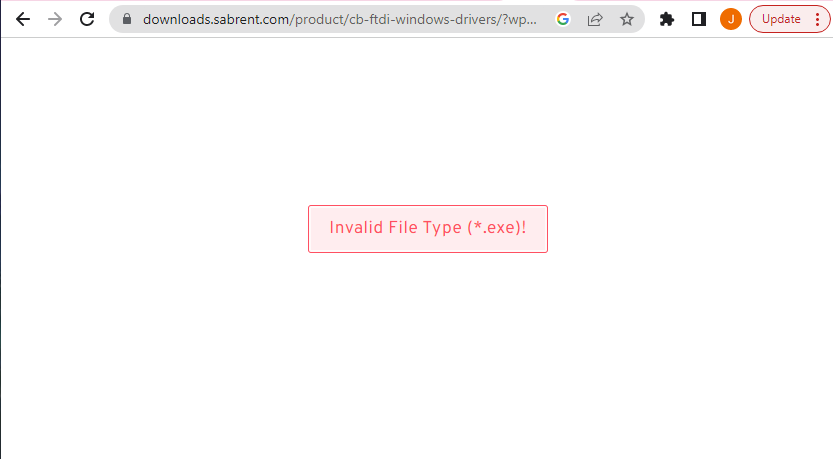
It is worth noting that each minor change requires a commit, and each commit takes many minutes. This small pause, over the course of many trials, collected to drag the process of this lab over several days. This gives more reason to incentivize better documentation and pre-planning, to reduce the amount of technical delay when implementing the lab.

In one of the attempts to initialize the tunnel, one classmate named *Derek Liu* helped us with the testing commands, *test vpn IPSec-sa* and *test vpn ike-sa*. In conjunction with monitoring the activity of the firewall on the System tab, it can be seen the vpn is attempting to initiate the gate, but nothing actually goes through so it fails, and deletes itself. Another suggestion we received was to make sure the IKE gateway was enabled. After double-checking and enabling all the gateways and interfaces that have the “enable” feature, nothing changed. The tunnel still did not work. It was later learned that if the configuration was properly set up, the tunnel will be established instantly. This means that the gateways are enabled upon creation and is the default. Thus, this means that the error was somewhere else and enabling is unnecessary.

As the lab took course over many days, there were issues with making sure the setup was identical every time. My partner, Evan Choi, and I are not the only ones to use the rack throughout the day. And so, as wires get unplugged and the PC’s addressing gets changed, it is easy to mess up the wiring and addressing topology every day. An incorrect connection would make problems appear consistent regardless of configuration changes. For example, we don’t know when, but one day we found that the ethernet 1/2, the outbound interface, was accidentally wired to the local interface of the opposing firewall. This topologically does not make sense and will not work. The outbound interface was corrected to point outwards (through the switch and router) to the opposing firewall’s outbound interface, which was also 1/1.

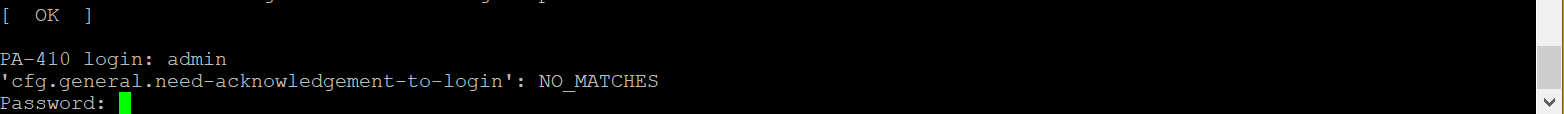
In terms of pinging knowledge, it is important to note that pinging the “default gateway” of the “LAN” from a PC in the trusted zone should not work. In other words, you cannot ping the firewall’s address by default.

After all of these changes and errors over such a long period of time, I decided to restart the entire lab from a factory reset. Also, it was intended to be done with 2 PA-220s instead of the two different firewalls. One of them had to be licensed up to date, however, so to save time, I brought one home, with the USB to Console adapter and a couple ethernet cables, so that I could update the licensing there. I assumed I needed the console adapter’s driver for it to work, but unfortunately the adapter’s website did not provide the driver.

 I tried to open PuTTY anyway. Using COM5 by checking the device manager, consoling worked without the driver. It is unsure why. After factory resetting and changing my PC’s USB to ethernet adapter’s addressing to be on the management interface, an address like 192.168.1.3 worked. I was then able to access the webGUI. It was very fun being able to access the firewall, which is typically in the lab environment, at home into my personal computer and router. However, the webGUI was noticeably slow. I then setup SOHO and DNS to retrieve licenses. SOHO worked, allowing me to plug my PC into my firewall for direct internet connection, which was very fun, but no matter which DNS servers I tried, it was not able to retrieve licenses. I gave up in attempting to license this PA-220 as during the same time two already-updated PA-410s are available for use back at the lab.

Unable to download driver for console to USB adapter.

With the new 410s, we also came with a completely new topology. We decided to follow the IP schematic of the Palo Alto’s suggested “quick config” for site-to-site VPN with static routing - <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>. With clearer private LAN and tunnel addressing, it is easier to recognize what interface or LAN is which. What was most prominent to note is that the addressing for step 2 on this official forum is incorrect. On step 2, number 6, the configuration for the tunnel interface should be the tunnel interface addresses they showed earlier in the topology, 10.10.10.10/24 and 10.10.10.11/24 (See *Lab Summary*).The tunnel interfaces should NOT point to LAN destination IPs. Finding an error in the quick configs of the official administrator page was greatly notable. Following the rest of the guide, with the proper interfaces to fit, it was only then the tunnel was established.

Here’s a quick mention of what the firewall configuration looks like if attempted to 

The last problem was about monitoring the session via a Cisco switch. Using WireShark, an external computer can see the packets that reach the connected port of the computer. However, we did not see the directed packets between each other, even after we inputted the commands. Turns out the way the command works is that the destination command “monitor session 1 destination interface f1/0/4” points to the interface that receives the copy of the packet, not the destination of the actual interface. So, the destination command should point to the switch interface of the computer with WireShark. The commands can be found here - <https://www.cisco.com/c/en/us/td/docs/routers/nfvis/switch‌_command/b-nfvis-switch-command-reference/monitor_commands.pdf>.

**Conclusion**

As the first VPN I had to set up, with many outside of school interruptions unto my lab schedule, countless recorded issues and failed attempts, and blatant hardware issues and delay, this lab took a long time. However, the ultimate configuration of this multi-layered process of setting up a VPN does show how complicated this technology could be with even further custom configuration and protocol. In its own right, Palo Alto’s VPN is still somewhat simple. Each process can be embedded to already existing SOHO configuration, and crypto key profiles and security policies can back off factory-reset defaults, such as the default virtual router and default security zone policies. Ultimately this lab taught me several fundamentals and greatly pushed my understanding of VPN technology in a future of cybersecurity.