PA-220 Firewall Factory Reset

Ryan Chen

Cybersecurity

Purpose:

The primary purpose of this lab is to bypass the password on an existing PA-220 firewall, given physical access. This allows us to familiarize with basic Palo Alto firewall access and control, such as interrupting the bootloader to access additional lower-level utilities.

Background Information:

Firewalls are networking devices primarily used to filter traffic coming in and out of a network. By nature, they are security devices, intended to protect users from malicious traffic. In this lab, we explore resetting the PA-220 firewall, a popular device manufactured by Palo Alto. Palo Alto is a company that specializes in cybersecurity and production of primarily network-based security equipment.

Due to the nature of this class, firewalls are shared amongst different years. Since passwords are set on previous year firewalls, they must be factory reset to be used once again. In the real world, this is the equivalent of providing used firewalls to a vendor. If they are not already reset, the passwords must be provided, or the operating system must be reset before being used.

To do so, we must interrupt the bootloader on the firewall to access more utilities, including the factory reset option. Palo Alto firewall maintenance options are obtained by searching up the options online, which Palo Alto provides documentation for.

The bootloader of any networking device such as a router or firewall is meant to initialize low level information necessary for the operating system to be loaded. This includes things like mapping the memory out, then loading the operating system in memory. Interrupting a bootloader indicates pausing the process of loading the operating system in, so we can access lower-level utilities.

Palo Alto provides written low-level utilities for such cases. In the event where a user might have forgotten their password or needs to recover the disk/alter the disk image, the Maintenance Recovery Tool allows them to perform these without booting into the actual operating system. This eliminates the need to enter the password, or boot into the OS (in the event where the OS may be corrupted). Other tools are provided as well, such as diagnostics and low-level utilities for OS recovery or scanning.

Palo Alto makes interrupting the bootloader simple by providing an interface for users to do so through a CLI. Users are required to type a sequence of text to stop the bootloader process and drop the firewall into a diagnostic state. This process will be demonstrated in this lab.

Lab Summary:

The first step to entering maintenance tools on the PA-220 is to interrupt the bootloader to boot into the utility. To do so, connect through console to the firewall port and power on the device. When the device is powered, text should appear documenting the process of startup.

```
SPI stage 1 bootloader
SPI ID: ef:40:18:00:00
header found at offset 0x1d80
Image 1.2: address: 0xffffffff81000000, header length: 192, data length: 8160
Validating data...
Starting next bootloader at 0xffffffff81000000
SPI stage 1.5 bootloader
SPI ID: ef:40:18:00:00
Header 1 found at offset 0x10000
Header 2 found at offset 0x100000
Found bootloaders, booting bootloader 1 of 2 at offset 0x10000.
Starting next bootloader at 0xffffffff81000000
Board type: KINGFISHER
    Welcome to the PanOS Bootloader.

U-Boot 11.0.0.0-63 (Build time: May 10 2023 - 21:55:50)

Octeon unique ID: 05c00040051df31e0850
N0.LMC0 Configuration Completed: 8192 MB
Warning: Board descriptor tuple not found in eeprom, using defaults
KINGFISHER board revision major:1, minor:0, serial #: unknown
OCTEON CN7130-AAP pass 1.2, Core clock: 1000 MHz, IO clock: 500 MHz, DDR clock: 800 MHz (1600 Mhz DDR)
Base DRAM address used by u-boot: 0x20fc00000, size: 0x400000
DRAM: 8 GiB
Clearing DRAM...... done
Using default environment

SF: Detected W25Q128BV with page size 256 Bytes, erase size 4 KiB, total 16 MiB
Found valid SPI bootloader at offset: 0x100000, size: 1310920 bytes
Loading bootloader from SPI offset 0x100000, size: 1310920 bytes

    Welcome to the PanOS Bootloader.

U-Boot 11.0.0.0-63 (Build time: May 10 2023 - 21:59:20)

Octeon unique ID: 05c00040051df31e0850
N0.LMC0 Configuration Completed: 8192 MB
KINGFISHER board revision major:1, minor:4, serial #: 012801110441
OCTEON CN7130-AAP pass 1.2, Core clock: 1000 MHz, IO clock: 500 MHz, DDR clock: 800 MHz (1600 Mhz DDR)
Base DRAM address used by u-boot: 0x20f000000, size: 0x1000000
DRAM: 8 GiB
Clearing DRAM...... done
Octeon MMC/SD0: 0
Using default environment

MMC:   Octeon MMC/SD0: 0, Octeon MMC/SD0: 0
Net:   octeth0, octeth1, octeth2, octeth3, octeth4, octeth5, octeth6, octeth7, octrgmii0 [PRIME]
```

The bootloader will prompt for user input. Typing “maint” will interrupt the bootloader and drop the firewall into the recovery tools.

```
    Welcome to the PanOS Bootloader.

U-Boot 11.0.0.0-63 (Build time: May 10 2023 - 21:59:20)

Octeon unique ID: 05c00040051df31e0850
N0.LMC0 Configuration Completed: 8192 MB
KINGFISHER board revision major:1, minor:4, serial #: 012801110441
OCTEON CN7130-AAP pass 1.2, Core clock: 1000 MHz, IO clock: 500 MHz, DDR clock: 800 MHz (1600 Mhz DDR)
Base DRAM address used by u-boot: 0x20f000000, size: 0x1000000
DRAM: 8 GiB
Clearing DRAM...... done
Octeon MMC/SD0: 0
Using default environment

MMC:   Octeon MMC/SD0: 0, Octeon MMC/SD0: 0
Net:   octeth0, octeth1, octeth2, octeth3, octeth4, octeth5, octeth6, octeth7, octrgmii0 [PRIME]
Type the command 'usb start' to scan for USB storage devices.



        Autoboot to default partition in 5 seconds.
        Enter 'maint' to boot to maint partition.

Entry: maint

Booting to maint mode.
```

Note that this process takes approximately 5-10 minutes depending on the speed of the firewall.

```
                    Welcome to the Maintenance Recovery Tool




     Welcome to maintenance mode. For support please contact Palo Alto
     Networks.

                 866-898-9087 or support@paloaltonetworks.com

< Continue                                                              >
```

When booted into the recovery tool, several options will appear.

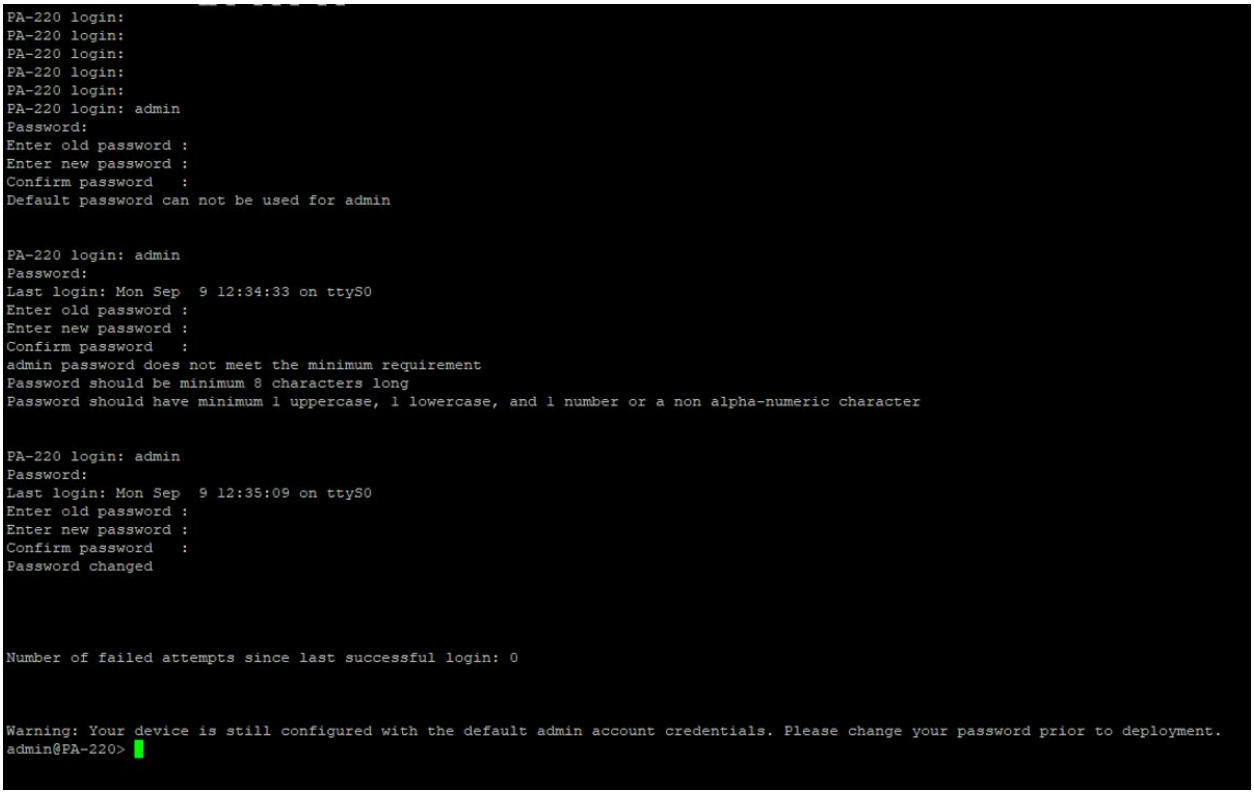
```
                    Welcome to the Maintenance Recovery Tool


< Maintenance Entry Reason                                          >
< Get System Info                                                   >
< Factory Reset                                                     >
< Set FIPS-CC Mode                                                  >
< FSCK (Disk Check)                                                 >
< Log Files                                                         >
< Bootloader Recovery                                               >
< Disk Image                                                        >
< Select Running Config                                             >
< Content Rollback                                                  >
< Set IP Address                                                    >
< Diagnostics                                                       >
< Debug Reboot                                                      >
< Reboot                                                            >



               Q=Quit,  Up/Down=Navigate,  ENTER=Select,  ESC=Back
```

To factory reset, select the corresponding tool, and make sure “Scrub” is deselected. The scrub tool is used to securely wipe the firewall by rewriting existing data. This process, although secure, takes a long time and is unnecessary in this case.

```
                                Factory Reset

WARNING: Performing a factory reset will remove all logs and configuration.

Using Image:
   (X) panos-10.2.6

WARNING: Scrubbing will iteratively write patterns on pancfg, panlogs, and any
extra disks to make retrieving the data more difficult.
NOTE: This could take several hours to several days if selected.  Scrubbing is
not recommended unless explicitly required.

    [ ] Scrub

If scrubbing, select scrub type:
   (X) nnsa                        ( ) dod

< Factory Reset                                                     >

< Advanced                                                          >



               Q=Quit,  Up/Down=Navigate,  ENTER=Select,  ESC=Back
```

When ready to reset, press enter and wait for the firewall to reset. There should be a progress bar to indicate the duration. After the firewall is done resetting, it should boot directly into login.

The login for a new firewall uses default credentials of “admin/admin”. After this, the firewall will prompt for a new password to be set, and the factory reset is completed.



Lab Commands:

The only new learned command in this lab is:

“maint” – (enter at bootloader menu)

Stands for "maintenance", used at the bootloader menu to enter Palo Alto maintenance tools.

Problems:

1) **Firewall Loading Time**
   The PA-220 firewall takes quite a long time to boot to anything, including OS/utilities. Our group was initially unaware of this, and thus we thought that we had a configuration issue with our console setup. After interrupting the bootloader, we saw no immediate change in startup. Instead, logging messages showed up indicating that the operating system was starting. We thought that this indicated a failure to interrupt the bootloader. However, we later realized that the firewall boot sequence takes a long time. After waiting a few minutes, the Maintenance Recovery Tool appeared, and we were able to proceed.

Conclusion:

This was an introductory lab required to proceed further with firewall configuration. Within this lab, skills necessary to reset the firewall were learned, such as interrupting the bootloader. This lab is also an introduction to CLI maintenance on Palo Alto firewalls, done through a console connection.

SOHO Palo Alto Configuration

Ryan Chen

Cybersecurity

Purpose:

The primary purpose of this lab is to setup a basic small office home office configuration suitable for the PA-220 firewall. This configuration involves setting appropriate VLANs, trust/security zones, and DHCP configurations in order to provide client machines with access to the internet. As a result, basic SOHO configuration provides an easy, gentle introduction to Palo Alto firewall utilities and functions.

Background Information:

Barebones configuration is commonly referred to as a SOHO or Small Office/Home Office setup. Traditionally, this means that the network will be used at a small scale for a limited number of clients. As a result, the config is robust at a small scale, and easy to setup. Despite these advantages, SOHO networks are often limited in future growth and must be upgraded to support anything larger scale.

Before utilizing a firewall, configuration must be done in order to prepare the device for use by clients. This involves setting up Dynamic Host Configuration Protocol (DHCP) to automatically provide client machines with IP addresses, which ensures that all devices can reach to the internet.

Our lab makes use of an upstream internet provider that already has DHCP leasing system set up for its clients. Thus, as a client, we can configure the firewall to receive a DHCP address on a specified interface, and then internally host a DHCP server to provide local clients with internet access. This would mimic a normal scenario where an ISP gives out a DHCP address for your upstream connection.

Leasing addresses is the act of giving out a specific IP address from a pool of addresses for clients connected. These addresses have a “lifespan” attribute attached to them, and after the time is up, they will need to renew another address. This ensures addresses are always moving in circulation, which is optimal in a network where devices are constantly connecting and disconnecting to the internet, such as a SOHO network.

To follow best practices, we must also adhere to basic security principals to ensure that traffic is tagged accordingly. Palo Alto provides Security Zones for this purpose. These zones are logical ways to group physical and virtual interfaces on the firewall and can be used for future monitoring and filtering. By default, traffic is restricted to within the zone, but can be configured using Security Policy Rules to allow multiple zones to communication with each other.

Within this lab, security zones and VLANs are used to separate trusted and untrusted interfaces/regions. Virtual Local Area Networks, or VLANs are used to separate physical interfaces through software, aka logical means. They operate on networking devices to

segment traffic based on the address number. This allows devices to communicate specifically with other devices in a defined address number. Devices that are not in the same VLAN must be configured in order to allow them to communicate with each other. The inbound interface that acts as a DHCP client is configured as Untrusted, since traffic is coming in from the internet. The port facing the LAN side that assigns DHCP addresses is Trusted since it is internal to the firewall/network.

Lab Summary:

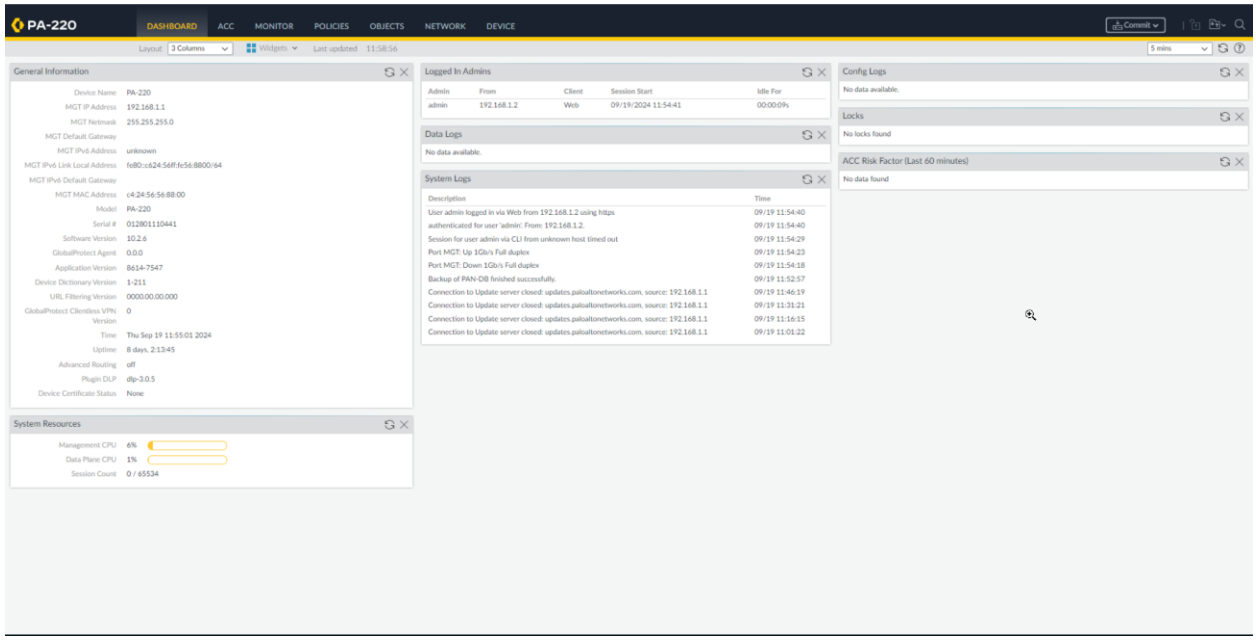
This lab involves a few relatively simple steps. First, DHCP is configured both inbound and outbound, providing internet connectivity to individual devices on the network. We can accomplished this by setting the outbound interface to act as a DHCP client and receive an address from an upstream DHCP provider (ISP). Then, we can configure the inbound interface to host a DHCP server that dynamically allocates addresses to all connected devices. Along this process, we should tag the interfaces using the correct VLANs and Security Zones for easier configuration in the future.

Once this is done, we should configure a Security Policy to allow traffic to flow through, which will establish internet connectivity if the DHCP server has already leased out the addresses. NAT should also be given a policy to allow the inner DHCP server addresses to be translated outward to the address given to the outbound port. This is crucial in allowing the clients to all reach the internet using one single ISP address.

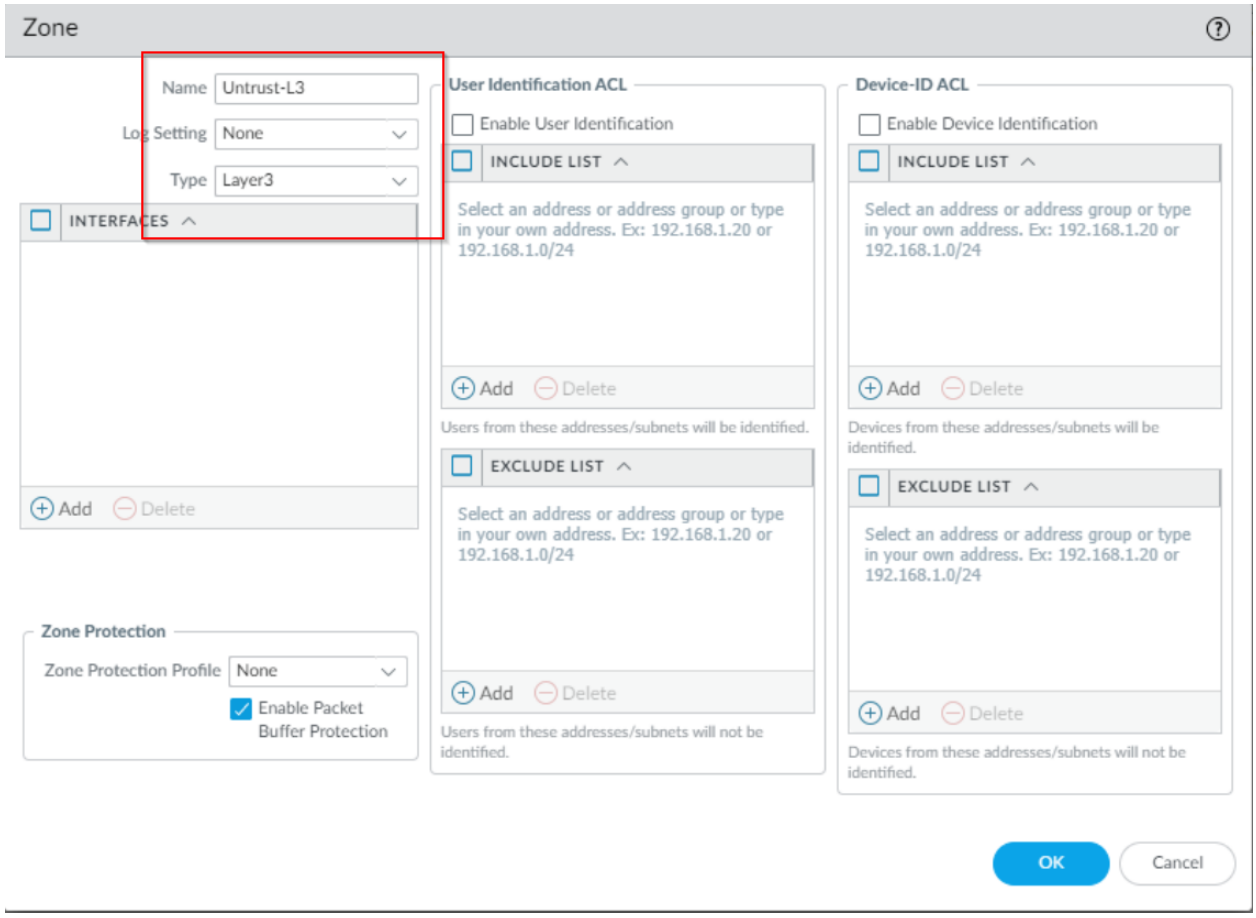
Finally, we can set the default management IP for future configuration and the default DNS servers to allow clients to automatically surf the internet. DNS is necessary so that domain names (such as https://google.com) is directly translated to the corresponding IP.

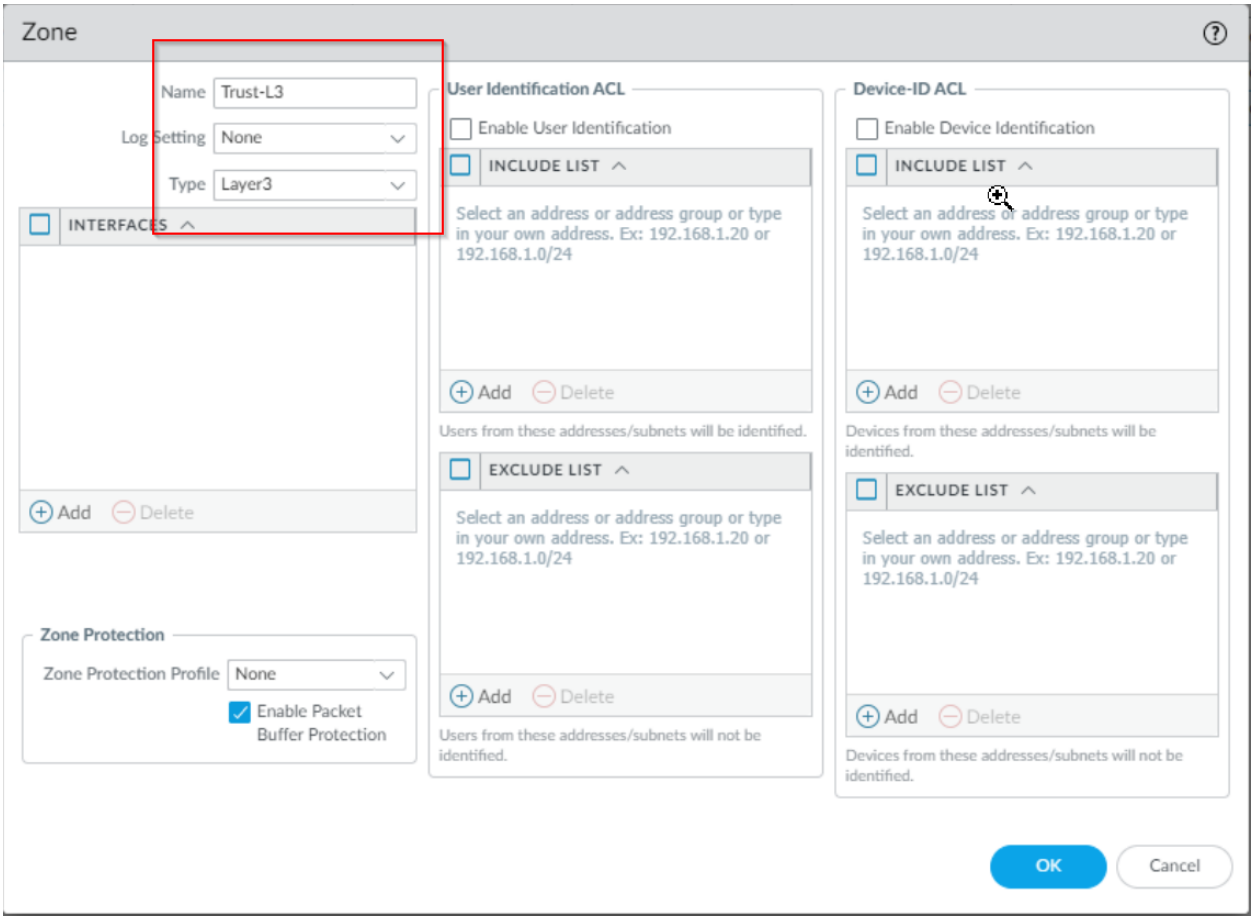
Lab Commands:

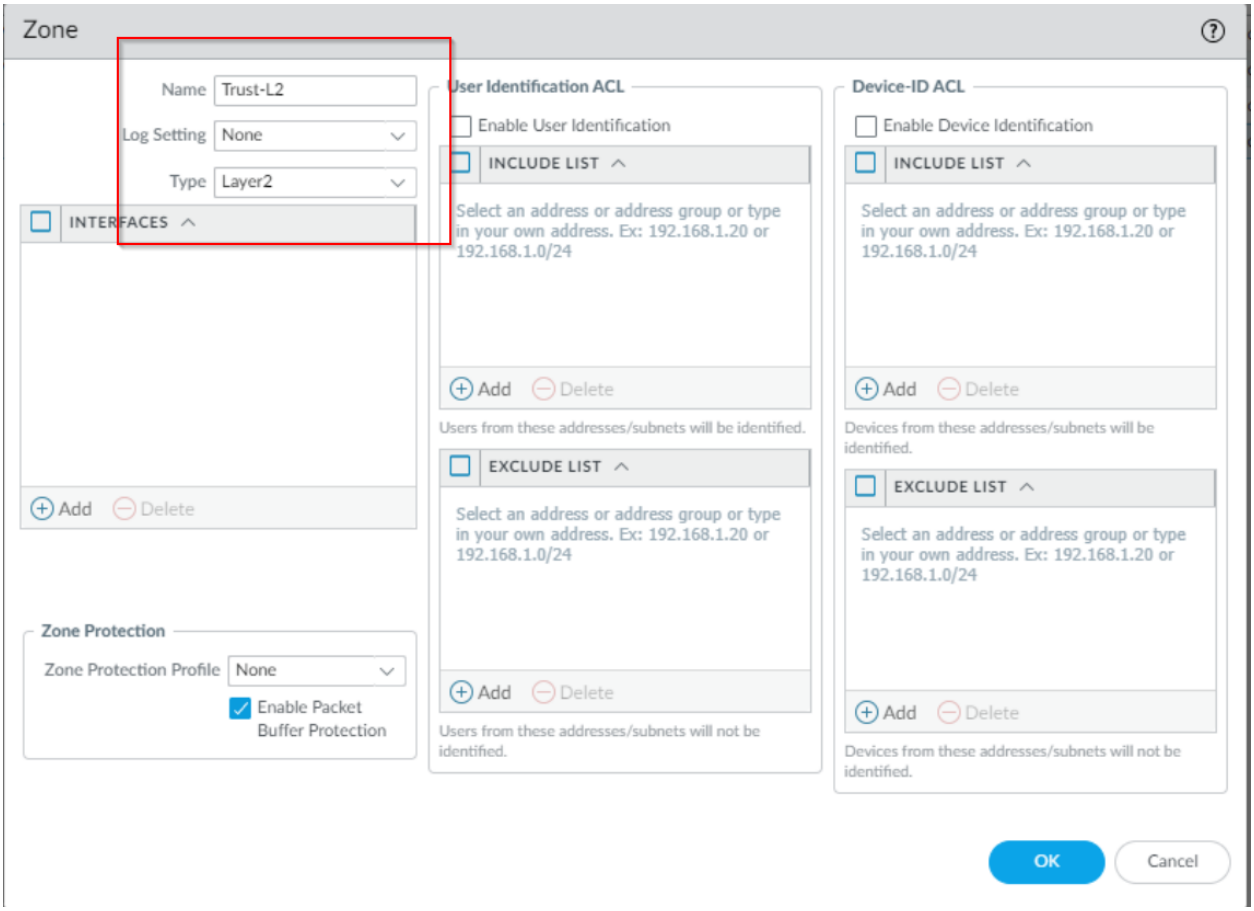
1) Ensure the interface is up and running, with nothing wrong on the dashboard. Make sure you can reach the firewall at the default address.



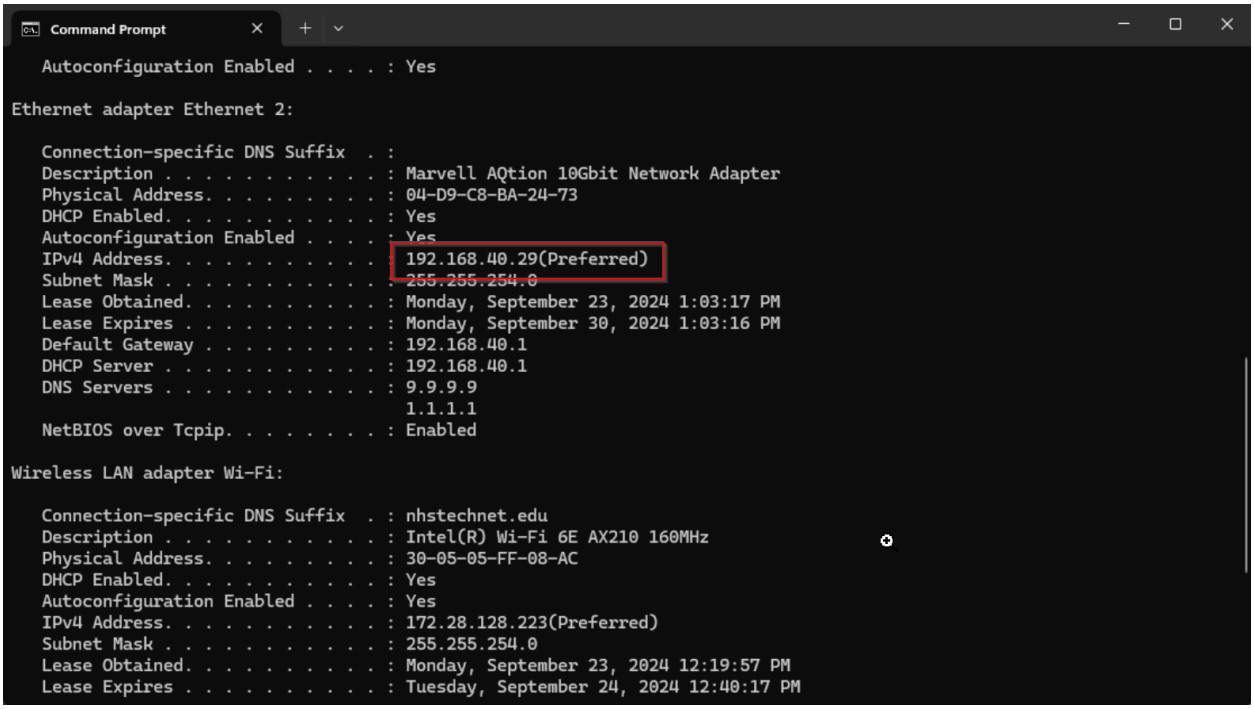
2) Make the Security Zones: Trust-L2, Trust-L3, Untrust-L3.


Zone
Name
Untrust-L3
Log Setting
None
Type
Layer3
INTERFACES
Add
Delete
Zone Protection
Zone Protection Profile
None
Enable Packet Buffer Protection
User Identification ACL
Enable User Identification
INCLUDE LIST
Select an address or address group or type in your own address. Ex: 192.168.1.20 or 192.168.1.0/24
Add
Delete
Users from these addresses/subnets will be identified.
EXCLUDE LIST
Select an address or address group or type in your own address. Ex: 192.168.1.20 or 192.168.1.0/24
Add
Delete
Users from these addresses/subnets will not be identified.
Device-ID ACL
Enable Device Identification
INCLUDE LIST
Select an address or address group or type in your own address. Ex: 192.168.1.20 or 192.168.1.0/24
Add
Delete
Devices from these addresses/subnets will be identified.
EXCLUDE LIST
Select an address or address group or type in your own address. Ex: 192.168.1.20 or 192.168.1.0/24
Add
Delete
Devices from these addresses/subnets will not be identified.
OK
Cancel

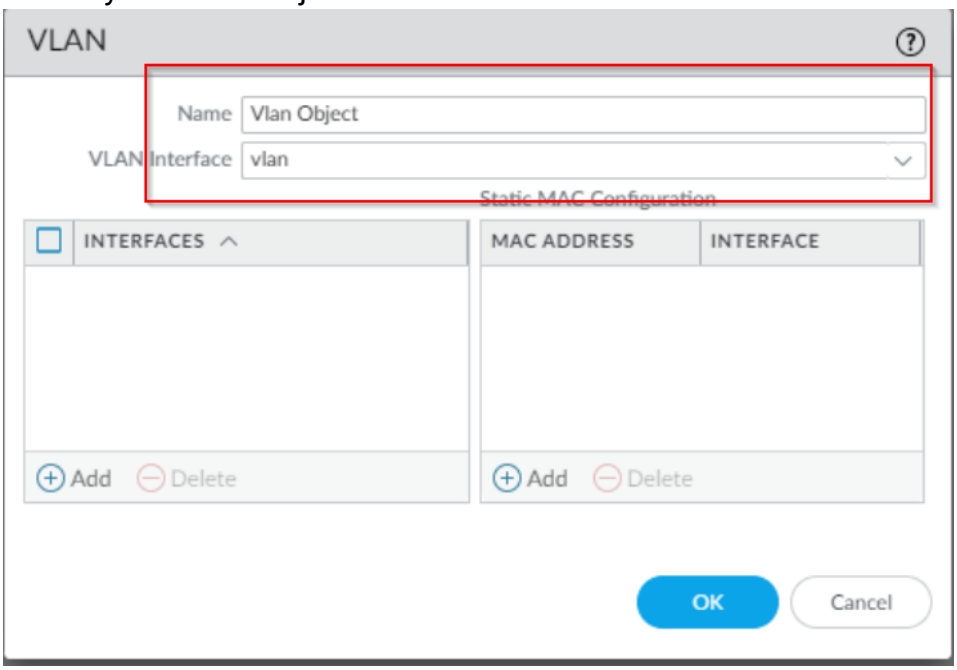

Zone
Name
Trust-L3
Log Setting
None
Type
Layer3
INTERFACES
Add
Delete
Zone Protection
Zone Protection Profile
None
Enable Packet Buffer Protection
User Identification ACL
Enable User Identification
INCLUDE LIST
Select an address or address group or type in your own address. Ex: 192.168.1.20 or 192.168.1.0/24
Add
Delete
Users from these addresses/subnets will be identified.
EXCLUDE LIST
Select an address or address group or type in your own address. Ex: 192.168.1.20 or 192.168.1.0/24
Add
Delete
Users from these addresses/subnets will not be identified.
Device-ID ACL
Enable Device Identification
INCLUDE LIST
Select an address or address group or type in your own address. Ex: 192.168.1.20 or 192.168.1.0/24
Add
Delete
Devices from these addresses/subnets will be identified.
EXCLUDE LIST
Select an address or address group or type in your own address. Ex: 192.168.1.20 or 192.168.1.0/24
Add
Delete
Devices from these addresses/subnets will not be identified.
OK
Cancel



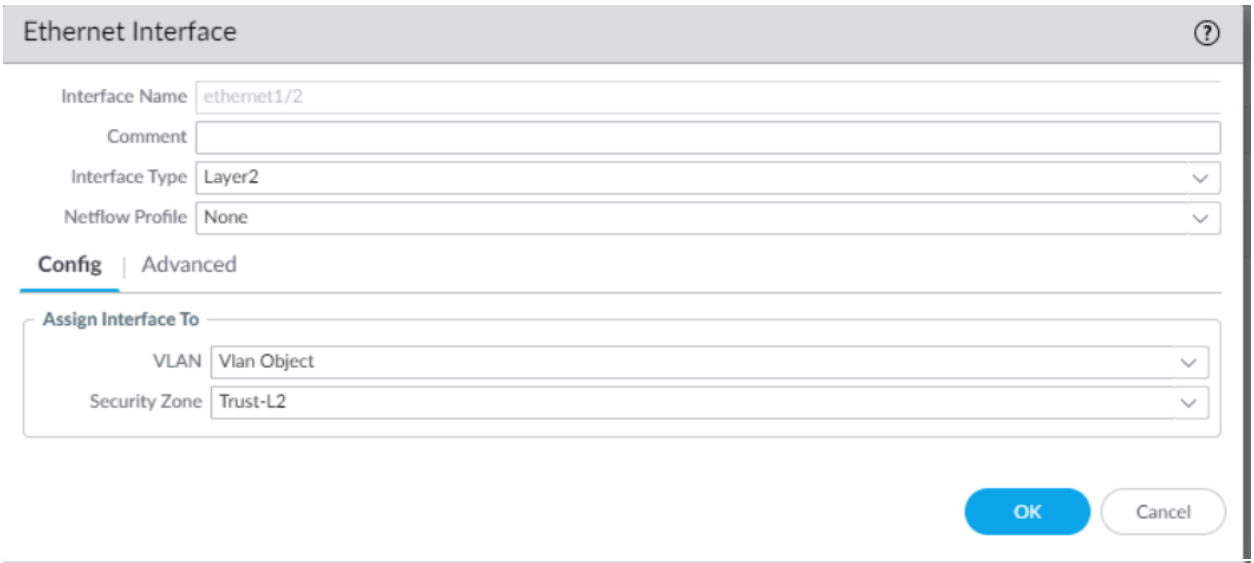
3) Before assigning an interface to DHCP, ensure that the upstream ISP is assigning an address. This is to reduce troubleshooting if configuration does not work on the firewall.

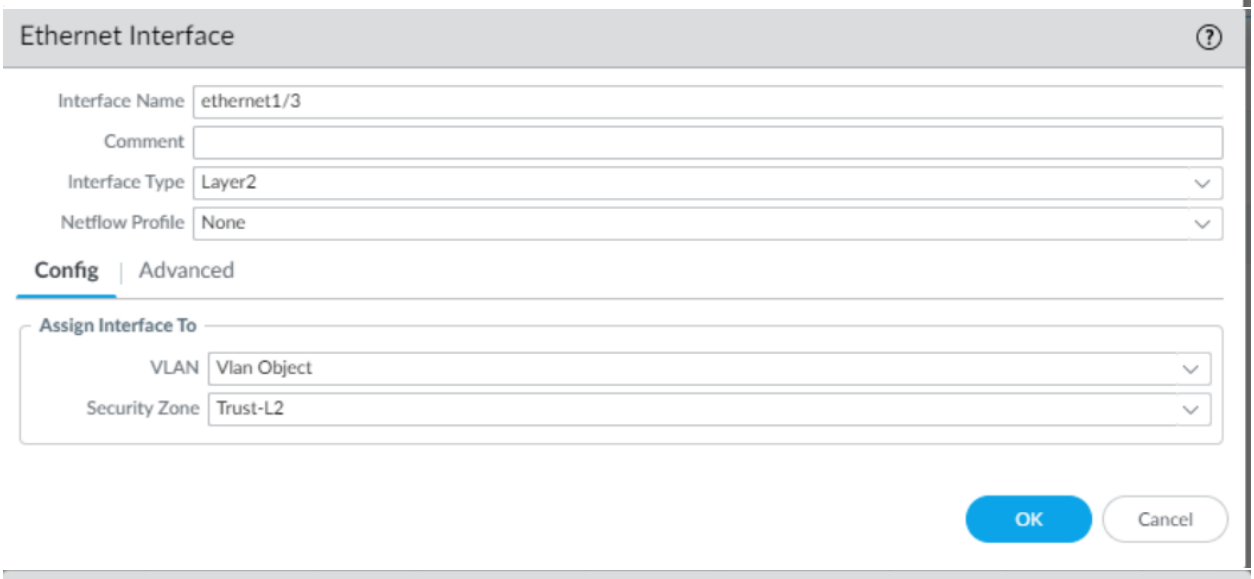


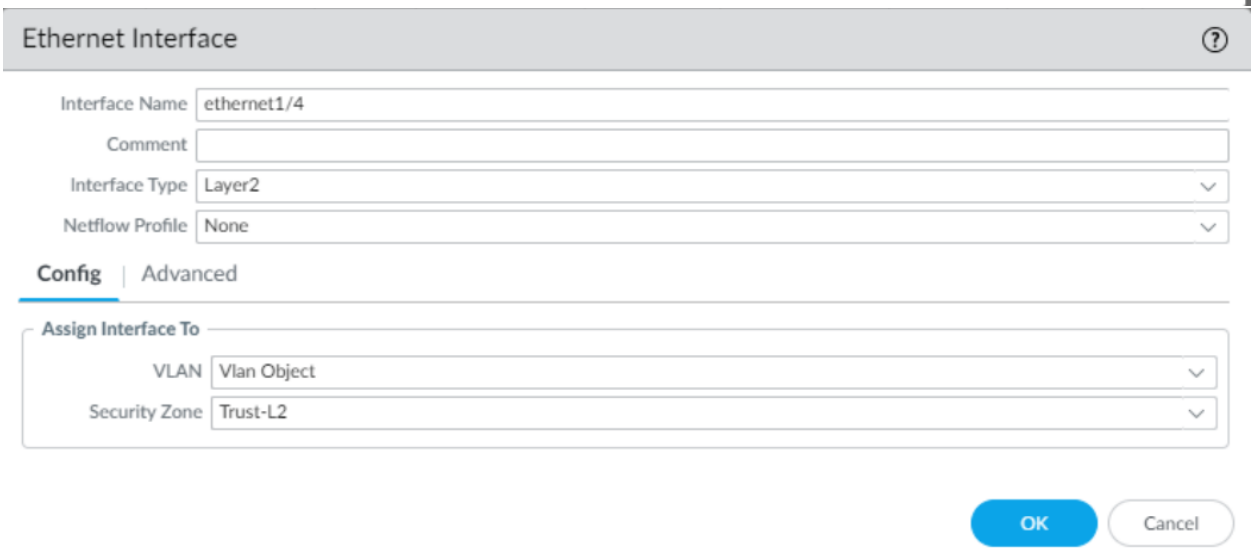
4) Create your VLAN object.



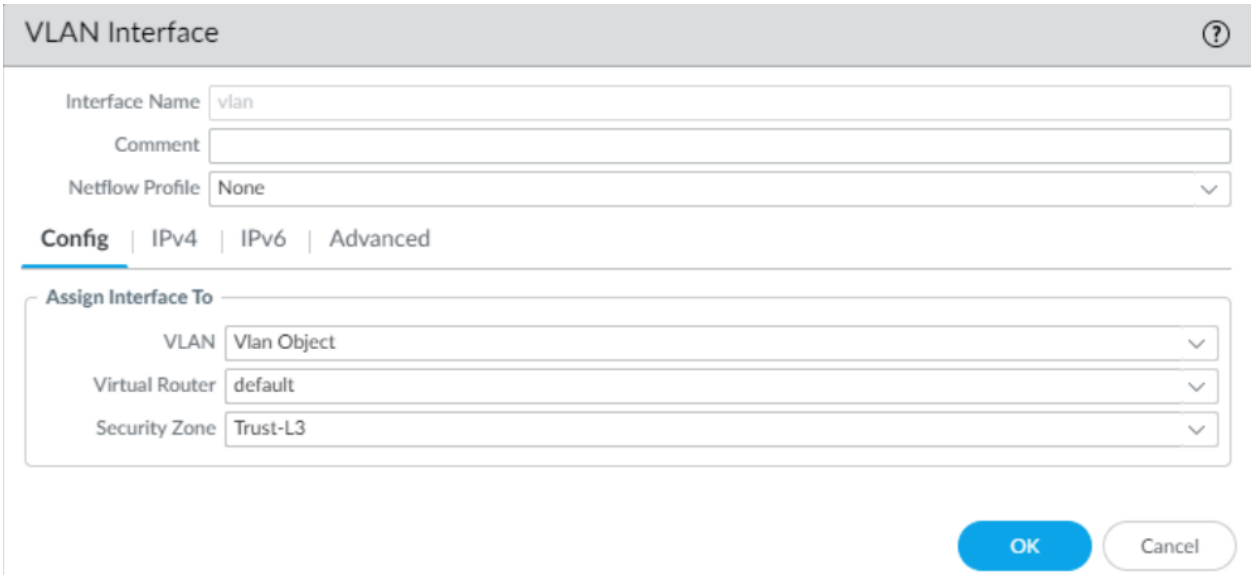
5) Configure your interfaces appropriately with the correct VLAN and Security Zone.

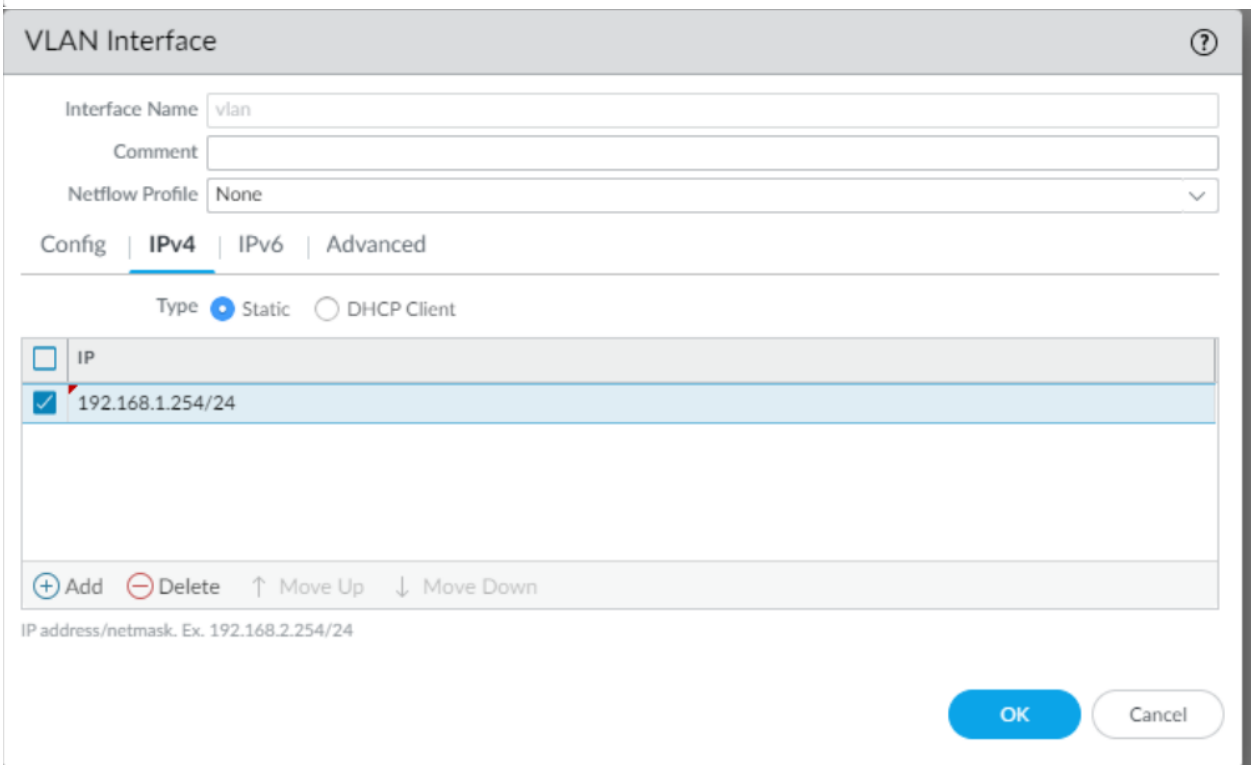




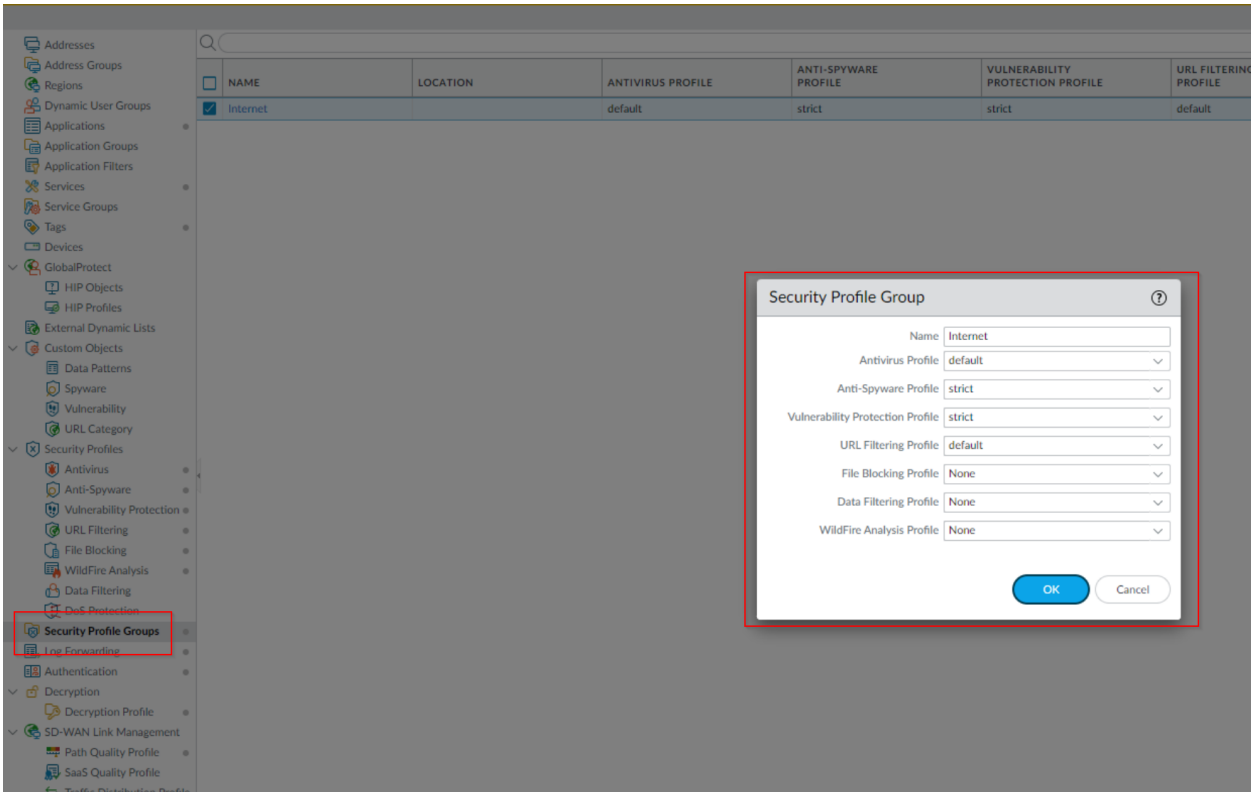


6) Configure the VLAN interface.

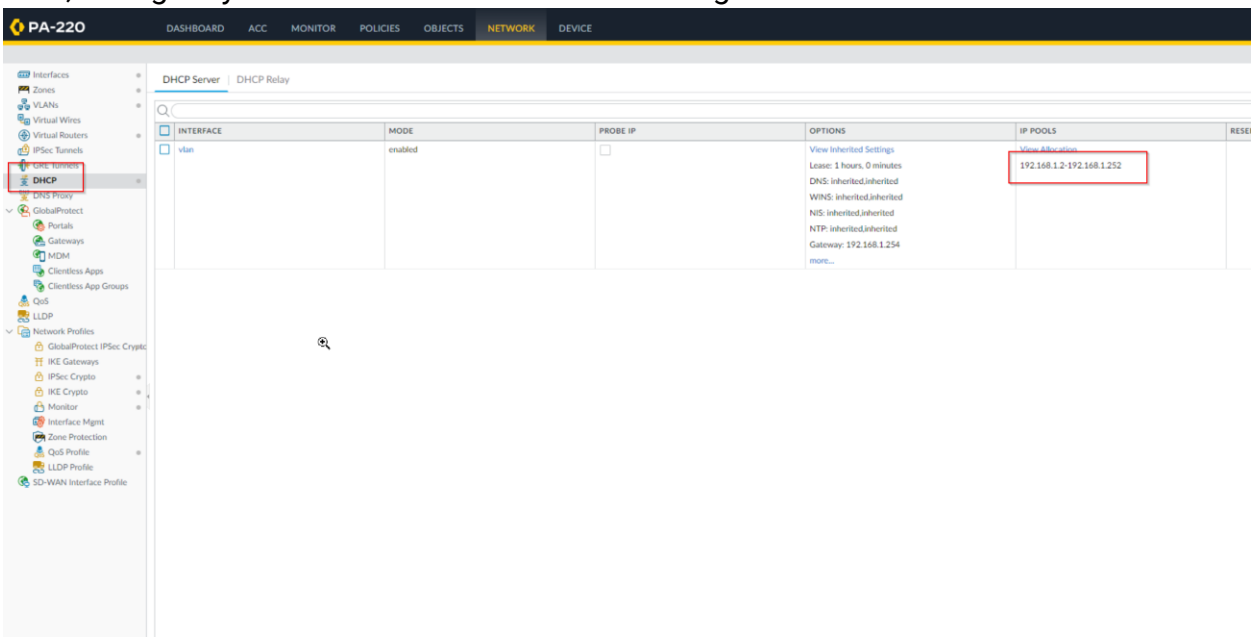


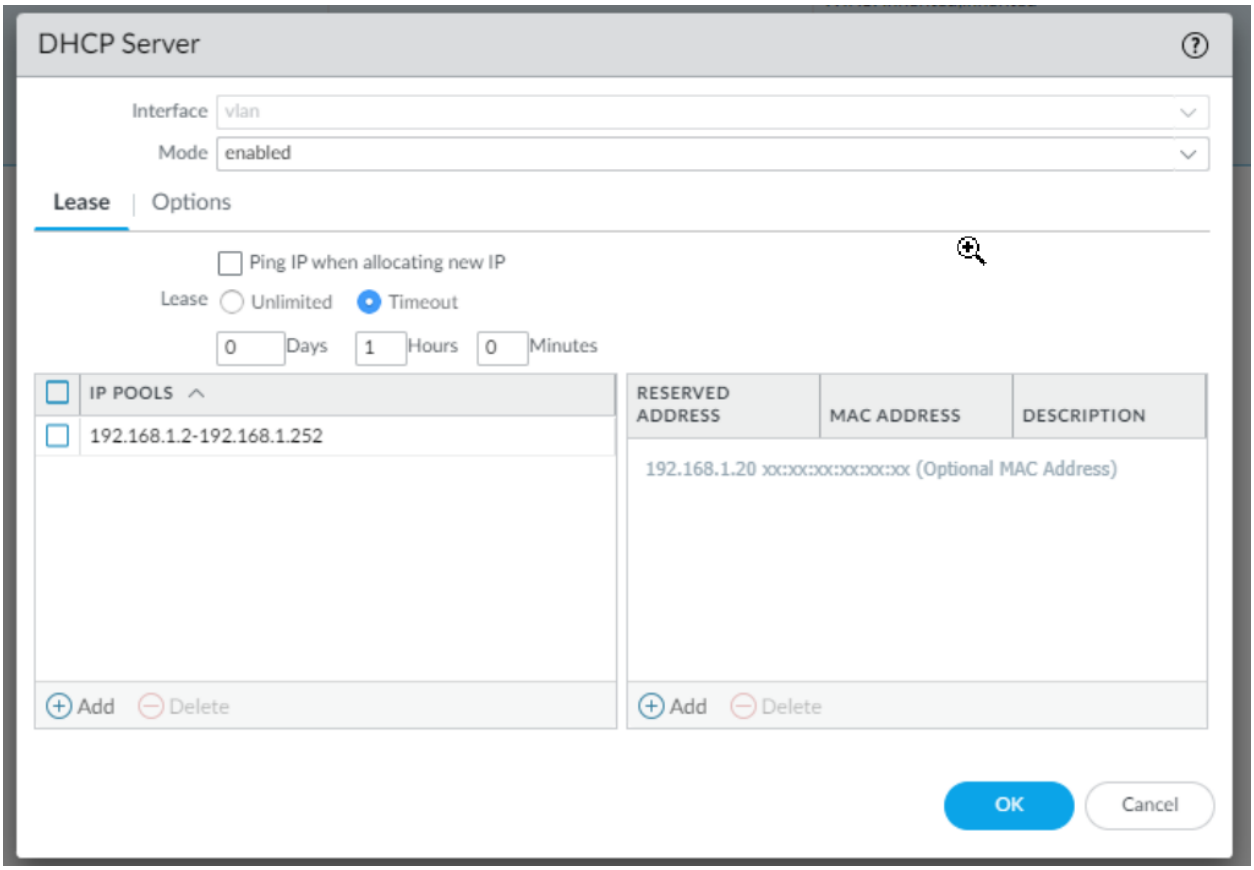


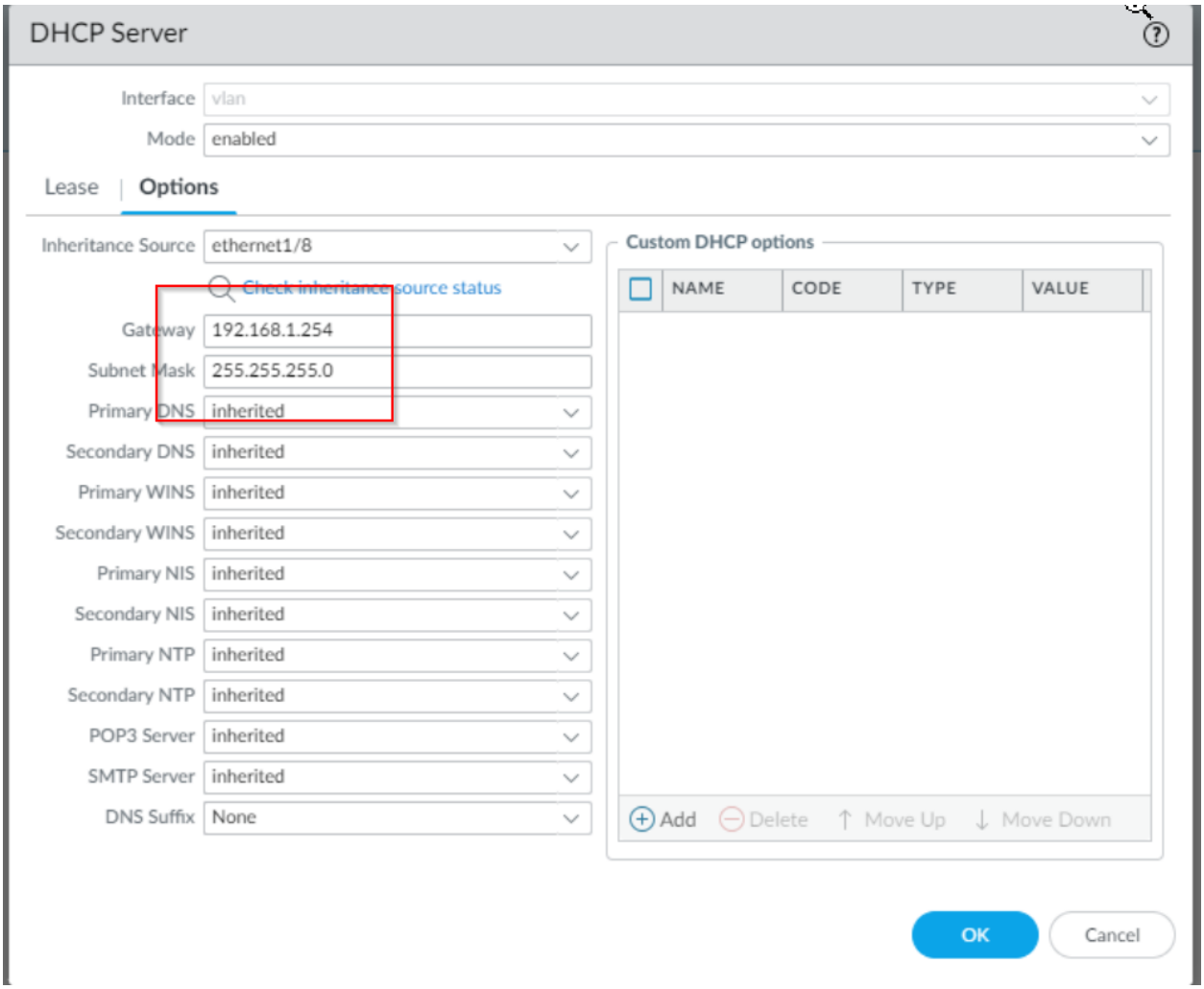
7) Assign your Security Profile group. Select the default.



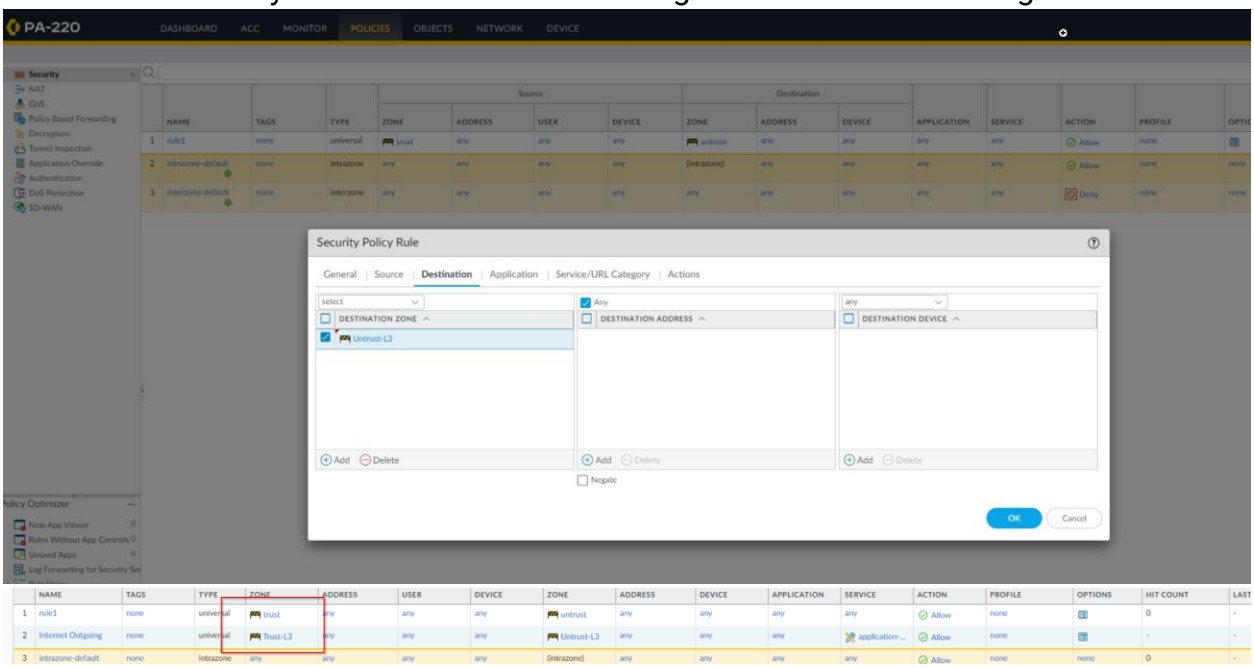
8) Now, configure your DHCP server with the following :



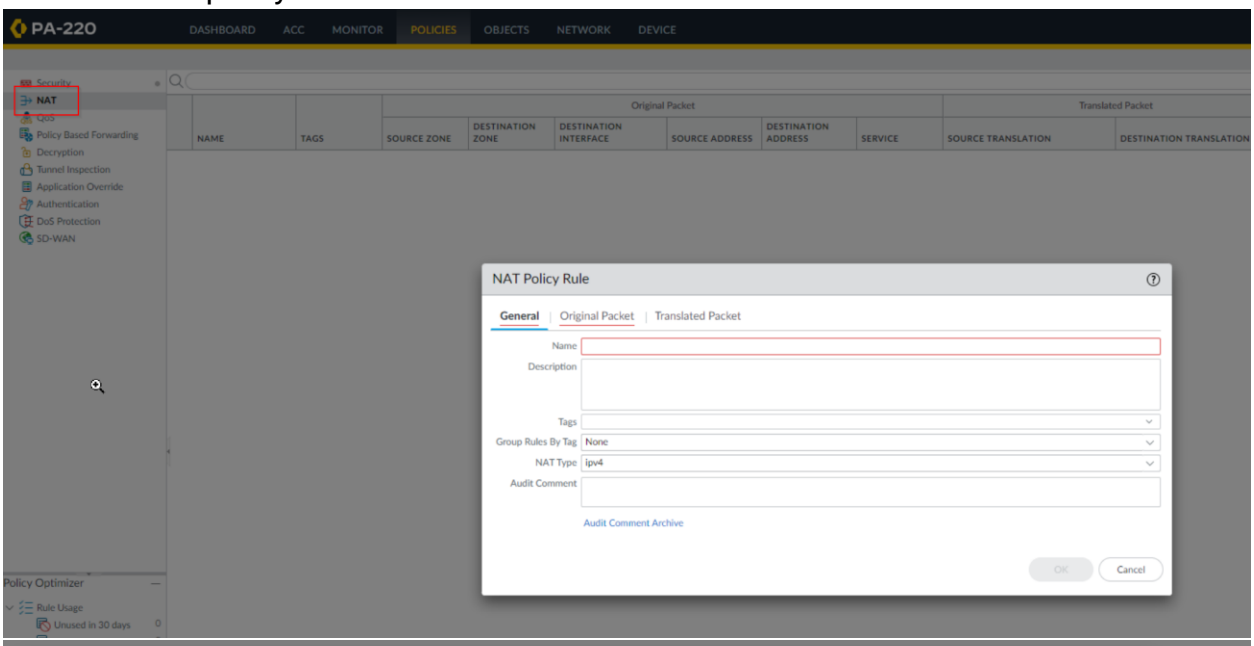

DHCP Server
Interface
vlan
Mode
enabled
Lease
Options
Ping IP when allocating new IP
Lease
Unlimited
Timeout
0
Days
1
Hours
0
Minutes
IP POOLS
192.168.1.2-192.168.1.252
Add
Delete
RESERVED ADDRESS
MAC ADDRESS
DESCRIPTION
192.168.1.20 xx:xx:xx:xx:xx:xx (Optional MAC Address)
Add
Delete
OK
Cancel

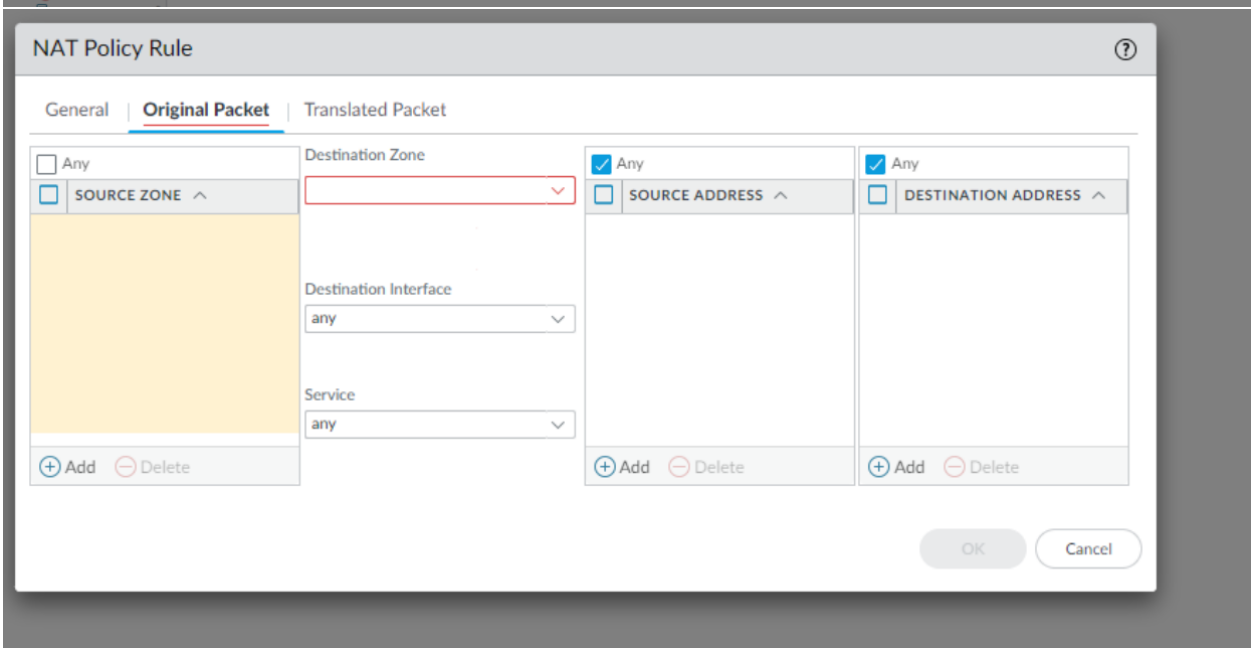


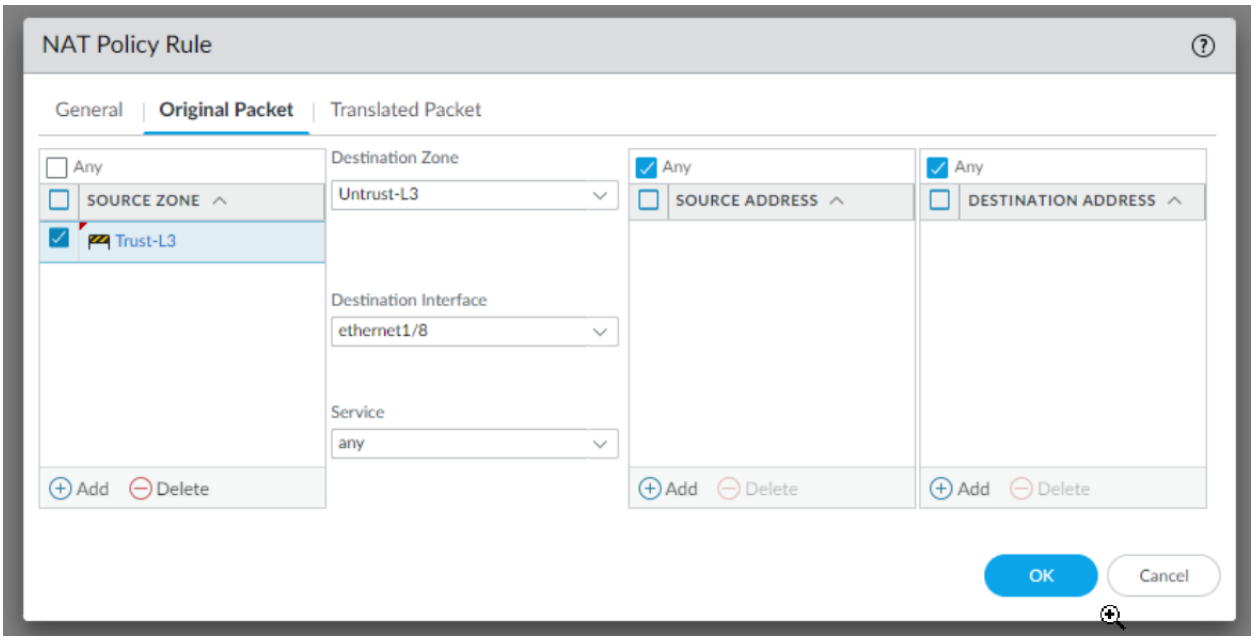
9) Create new Security Policies to allow Untrust regions to send to Trust regions.

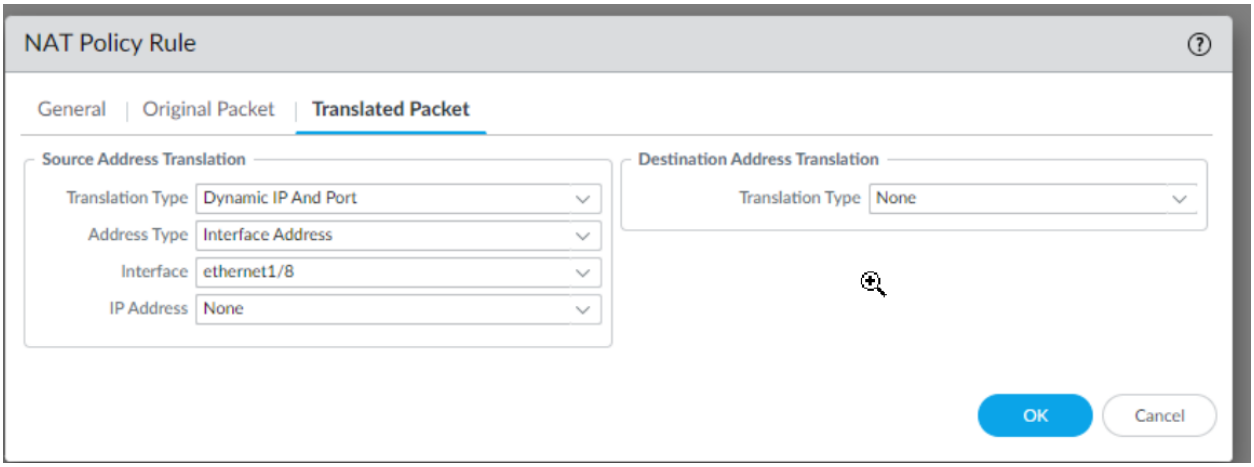


10) Create a NAT policy to enable DHCP clients to reach the internet.

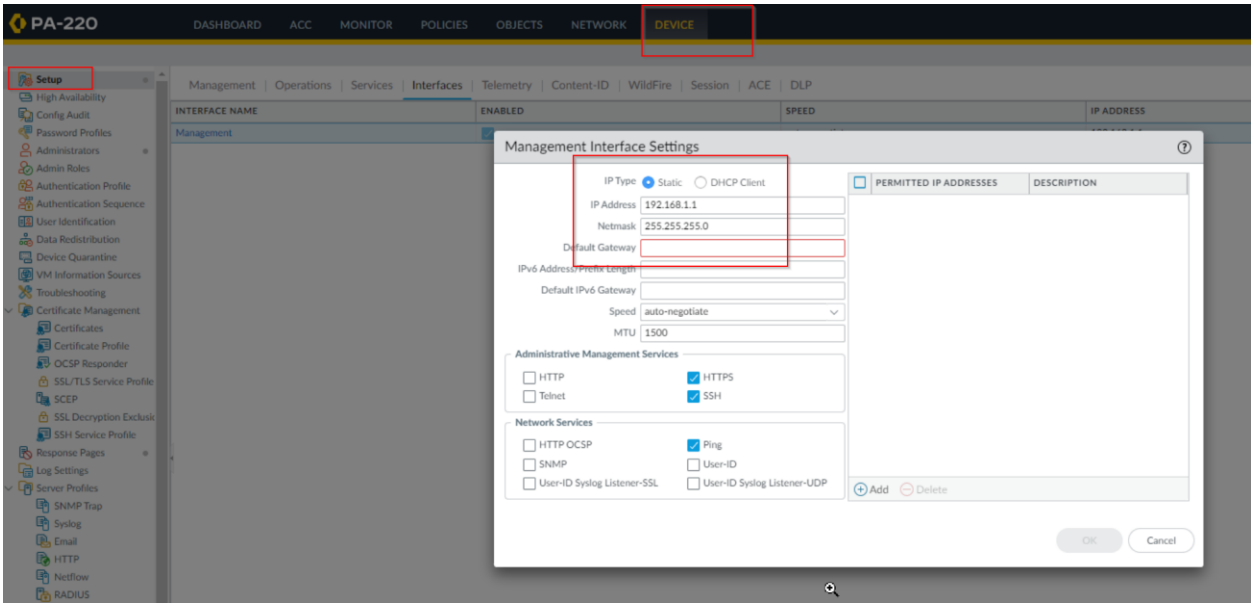




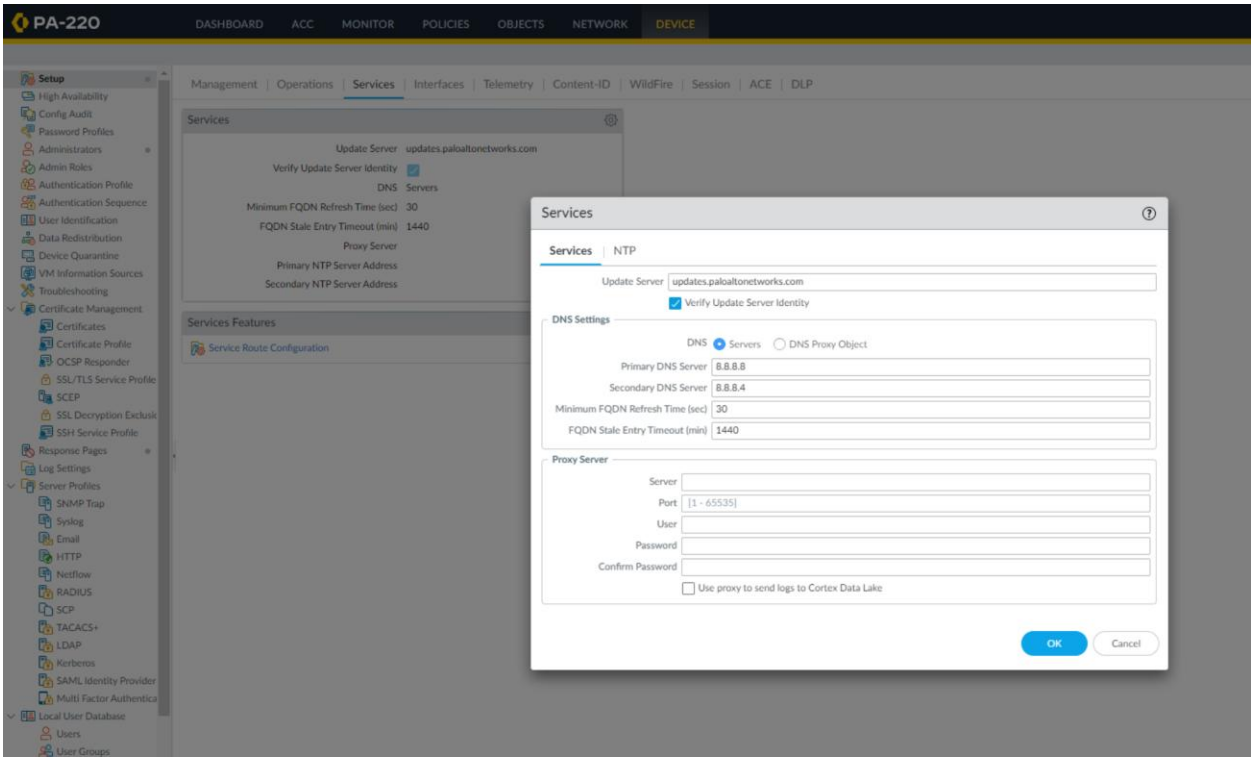




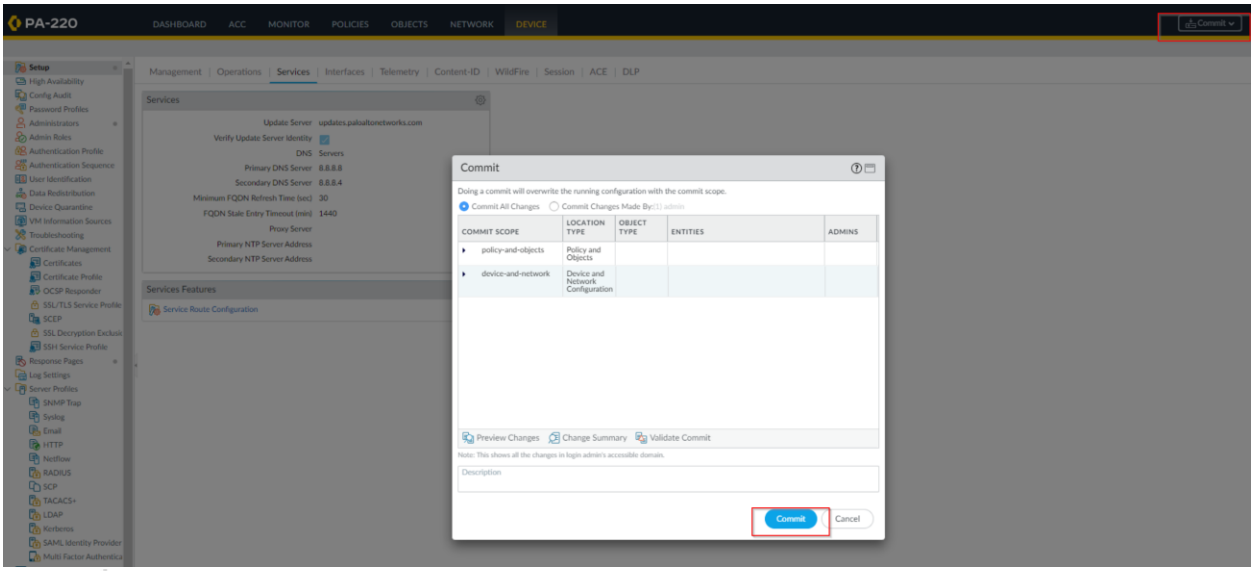
11) Finally, set the default management IP information so that the firewall can be accessed on the new network.

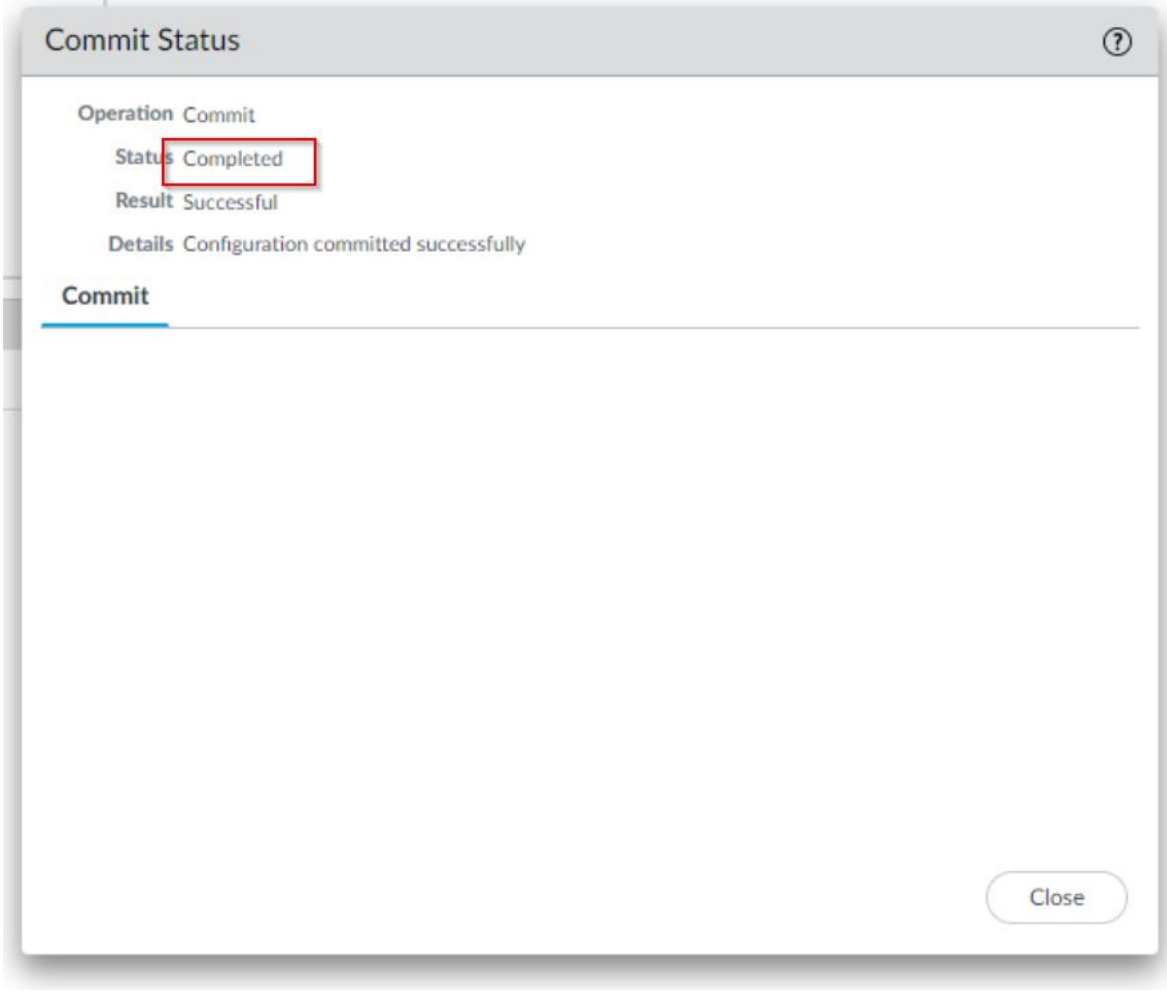


12) Set the DNS server.



13) Finally, commit the changes to ensure that it is saved to the firewall. After the firewall loads, all configurations should be applied.


PA-220
DASHBOARD
ACC
MONITOR
POLICIES
OBJECTS
NETWORK
DEVICE
Commit
Management
Operations
Services
Interfaces
Telemetry
Content-ID
WildFire
Session
ACE
DLP
Services
Update Server updates.paloaltonetworks.com
Verify Update Server Identity
DNS Servers
Primary DNS Server 8.8.8.8
Secondary DNS Server 8.8.8.4
Minimum FQDN Refresh Time (sec) 30
FQDN Stale Entry Timeout (min) 1440
Proxy Server
Primary NTP Server Address
Secondary NTP Server Address
Services Features
Service Route Configuration
Commit
Doing a commit will overwrite the running configuration with the commit scope.
Commit All Changes
Commit Changes Made By:(1) admin
COMMIT SCOPE
LOCATION TYPE
OBJECT TYPE
ENTITIES
ADMINS
policy-and-objects
Policy and Objects
device-and-network
Device and Network Configuration
Preview Changes
Change Summary
Validate Commit
Note: This shows all the changes in login admin's accessible domains.
Description
Commit
Cancel


Commit Status
Operation Commit
Status Completed
Result Successful
Details Configuration committed successfully
Commit
Close

Problems:

2) **“Invalid interface ethernet0/x”**
   We faced an odd issue involving DHCP configuration on the individual interfaces. At first, when we followed steps to assign an interface to be a DHCP client, it would not display any lease information. When we tried to renew or release DHCP, a generic error message popped up, stating that the interface was “invalid”. After a lot of trial error and several firewall resets, we realized that this was due to the default virtual wire not being deleted. After deleting the virtual wire and committing changes, we were able to successfully obtain a DHCP lease from upstream.
3) **Firewall Loading Time**
   Palo Alto firewalls take quite a while to load all utilities. At first, we did not realize this, and mistakenly thought that our firewall was bugged due to an inability to reach the web interface through the default IP. However, after waiting a few more minutes, all lights on the front of the firewall turned green. The web interface is then available through the default IP 192.168.1.1.

Conclusion:

Overall, this lab was a good introduction to what functions Palo Alto firewalls have, what specific issues might arise when configuring them, and how to resolve said issues. Through the configuration, we learned how a basic SOHO firewall configuration functions and how it is structured, as well as how to configure it.

URL Filtering on Palo Alto PA-220

Ryan Chen

Cybersecurity

Purpose:

The primary purpose of this lab is to configure URL Filtering for end-devices on a generic Palo Alto device. In this case, we are using the PA-220 firewall. This configuration involves creating a URL Filtering object, changing appropriate fields to block or unblock, and allowing end-users to override filtering with a password. To do so, root SSL certificates must also be installed on end-devices in order to ensure HTTPS websites can be intercepted to display the block page.

Background Information:

URL Filtering is a function on most firewalls (Palo Alto, Fortinet, Juniper, etc.) that allows administrators to configure certain websites to either allow, block, or require a log-in/override page to access. This kind of functionality is desirable for many reasons, in all kinds of environments.

For instance, administrators of work environments may need URL Filtering to block certain game, adult, or phishing websites on employee end-devices to increase both productivity and security. By restricting employee access to potentially dangerous websites, security is hardened through risk avoidance. Alternatively, a school administrator would be likely to block the same categories, including other additional ones like AI websites and social media sites.

Palo Alto implements URL Filtering through their custom block lists. All sites are added by Palo Alto internally into their own categories, and administrators can also create or modify their own lists. Predefined categories can be used with a URL Filtering license, and are listed here. Furthermore, Palo Alto provides a website where URLs can be checked for their categorization on their servers, which can be found here. This is useful for testing sites prior to categorization to ensure correct filtering behavior.

In order to configure URL Filtering on Palo Alto devices, an object must first be created to define the filtering profile. With a valid filtering license, Palo Alto's predefined categories can be configured to either allow, block, or override. Overriding is a feature where end-users can bypass URL filters temporarily by entering credentials. This is especially useful for cases where end-users may temporarily need to access a blocked website, such as AI. Timers are generally attached to these override functions, granting permissions to the site for a predetermined amount of time. After creating the profile, it can be applied in the policies menu, allowing filtering to be enabled.

Notably, one of the biggest issues with URL Filtering is the handling of HTTP vs HTTPS websites. Hypertext Transfer Protocol, or HTTP is a protocol that handles the core foundation of loading webpages. HTTP requests fetch data from a website, which is then sent to client machines and allows a website to be loaded. The biggest difference

between HTTP and HTTPS is security and encryption. HTTP Secure, or HTTPS has been widely adopted ever since the late 90's, as computer security has become more important. Because of HTTP's unencrypted nature, numerous situations may arise that allow attackers to intercept data, leaking important data. On the other hand, HTTPS utilizes SSL, or more recently TLS to encrypt requests and responses. TLS implements public and private key cryptography to establish a secure connection and send encrypted data.

The nature of block/filter pages requires the firewall to intercept the request and insert the block page in the place of the response. With HTTP, this is fairly easy to do, however HTTPS renders this impossible (by design) without necessary certificates. SSL/TLS certificates store the previously mentioned public keys for each website, along with other necessary information to establish a connection. Certificates are typically issued by certificate authorities, companies that generate certificates for people hosting websites. Then, these companies root certificates (the base, most trusted certificate) are pre-installed to popular operating systems like Windows, Mac, and Linux. Root certificates are trusted by default and used to verify the authority of other certificates for individual websites. Intercepting the request and inserting data using the firewall is impossible because we would need to decrypt and verify the data using certificates that we do not have access to. As a result, we need to provide our own self-signed root certificate, making our firewall trusted by end-devices. This will allow our connection to go through and show the block page, but it will unfortunately still show that our connection is "Insecure", due to it being self-signed.

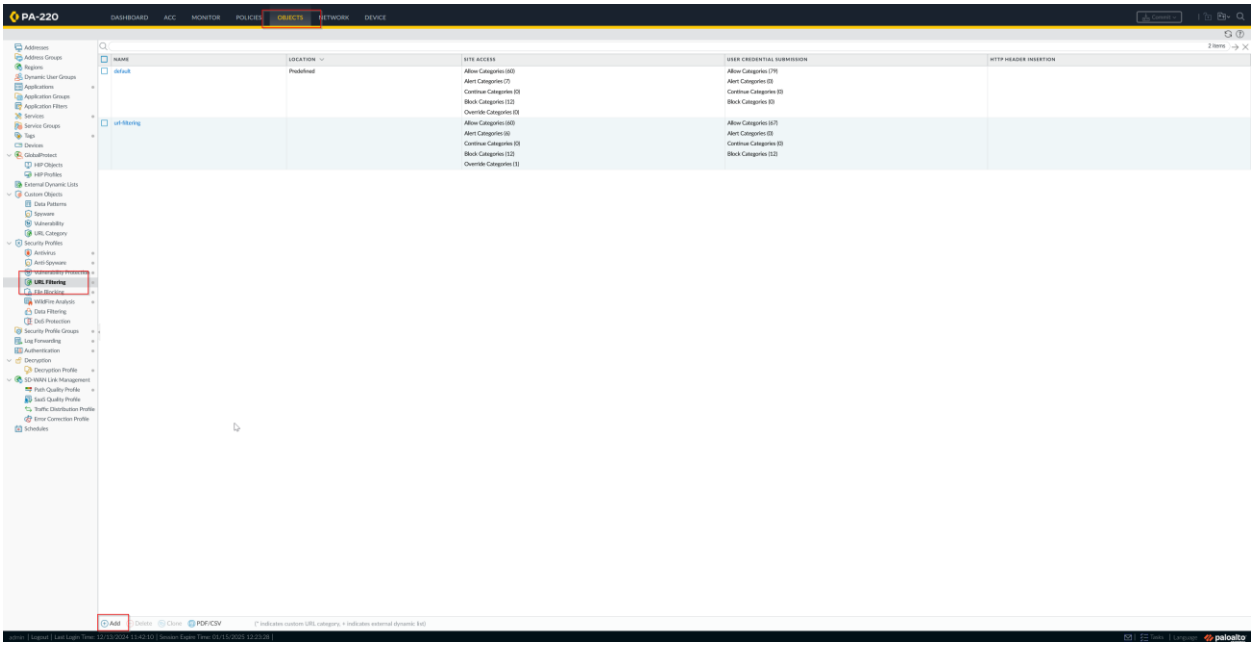
Lab Summary:

This lab involves a few separate steps. First, we must create the security object for URL Filtering, which we will later direct our firewall to use. By setting certain predefined Palo Alto categories to allow/block/override, we can grant permissions associated to certain websites. After the object is created, we can use it by setting it to apply to all outbound traffic. This will allow us to filter HTTP traffic.

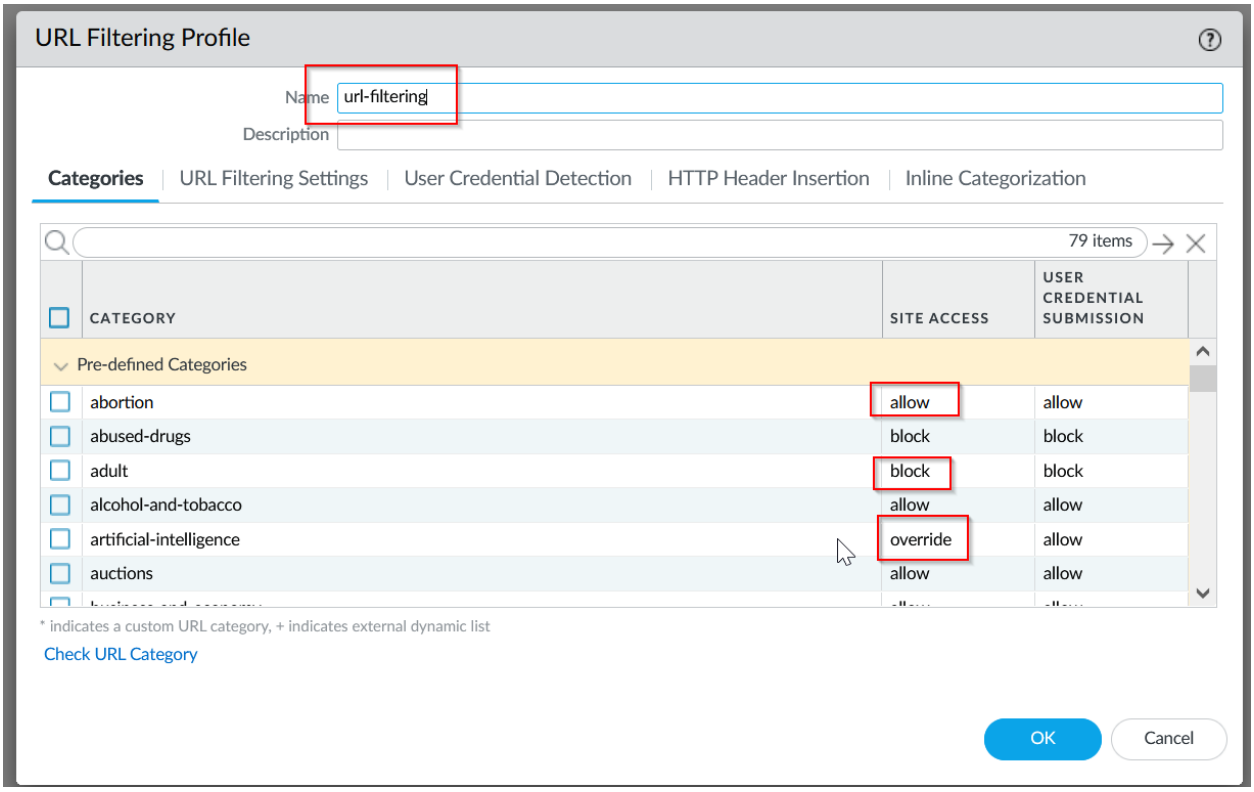
In order to filter HTTPS traffic and enable override pages, we must configure SSL/TLS decryption by generating our own root certificate and installing it into end-device machines. This allows end-devices to trust our firewall, giving us permission to intercept the traffic and instead replace it with the block/override page. Finally, override pages can be configured with specific passwords that end-users can input if they want to access the webpage.

Lab Commands:

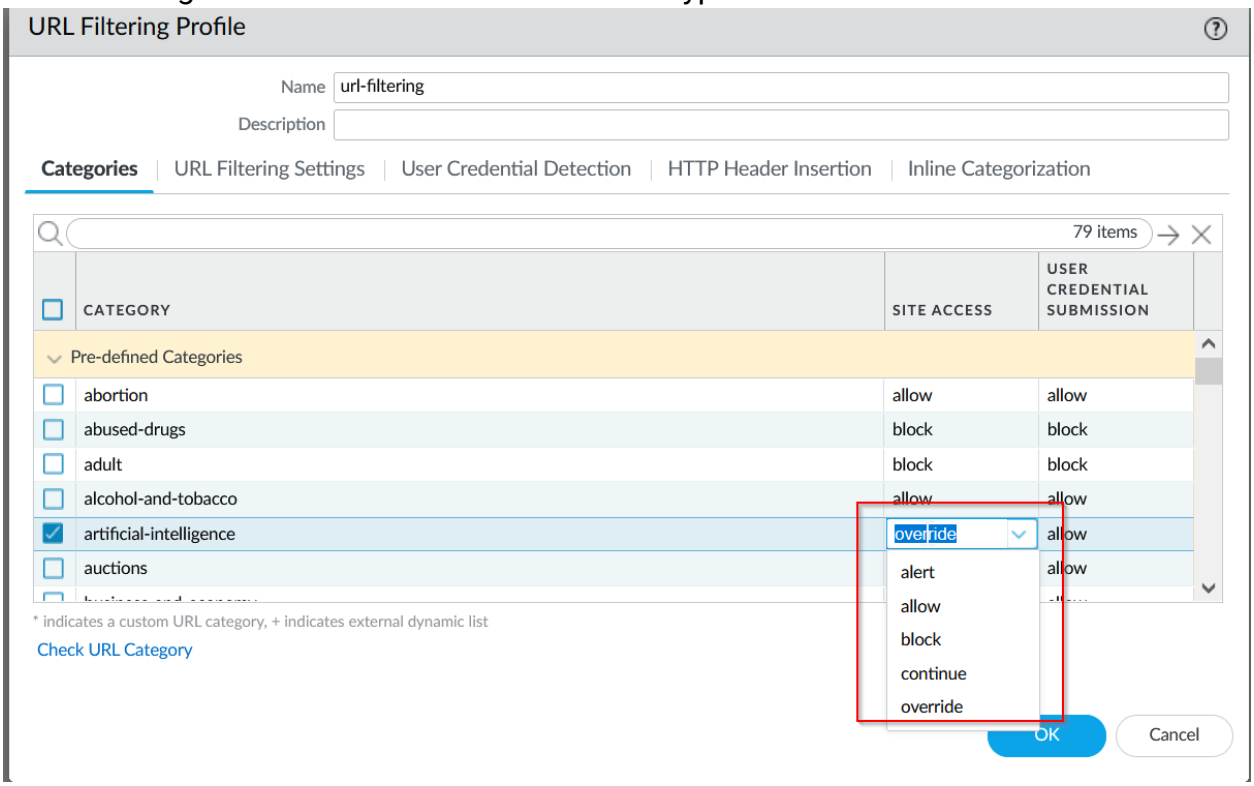
1) Navigate to Objects > Security Profiles > URL Filtering and press Add to create a new URL Filtering profile.



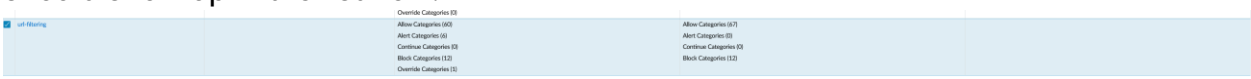
2) Here, you can name the profile and change the responses to certain predefined categories provided by Palo Alto.



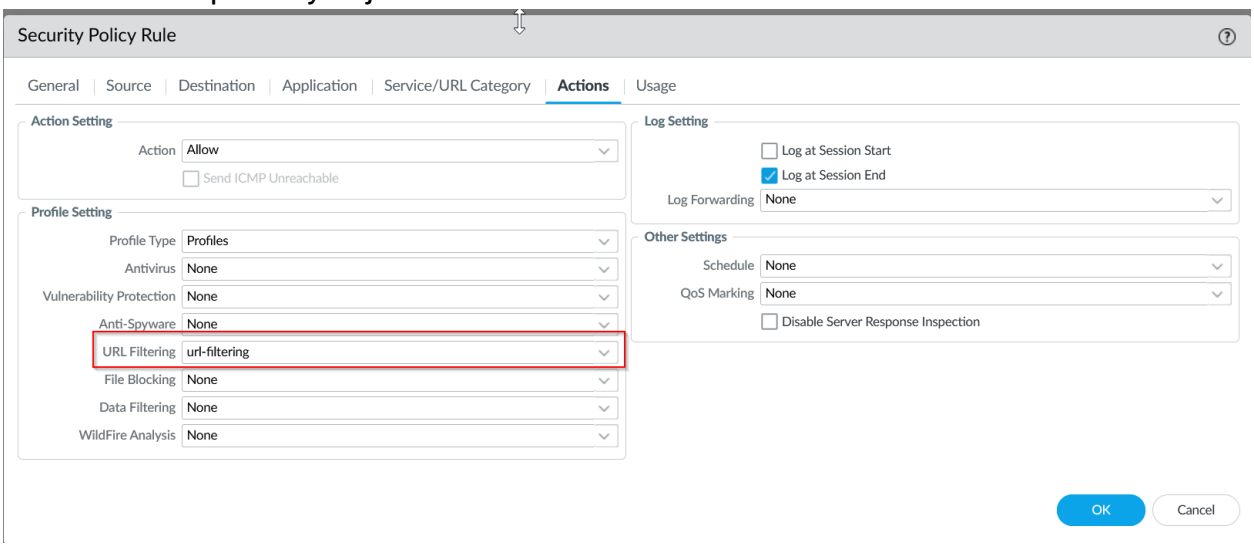
3) For later configuration, to make a category bypass-able with passwords, you should configure it as an “override” site access type.



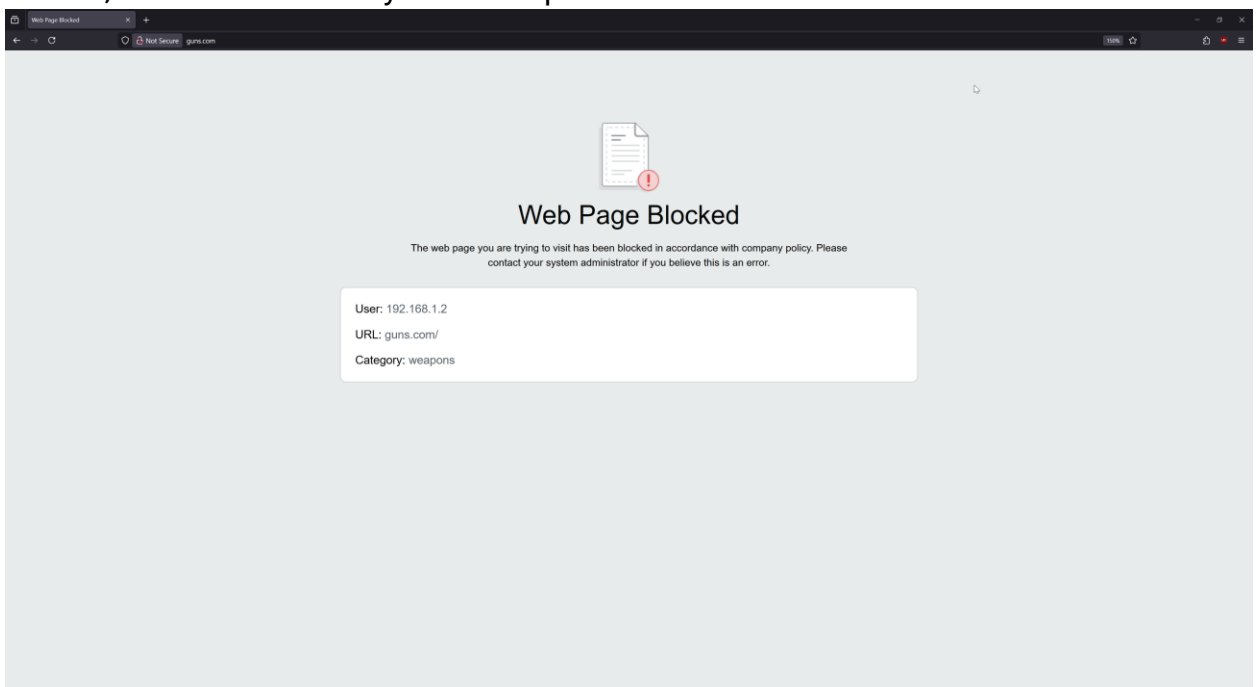
4) Keep the other settings the same and create the profile by pressing “Ok”. It should show up in the list now!



5) Apply the policy by going to Policies > Security now. We will apply the profile to the “outgoing” security policy to filter out websites. Go to the “Actions” section and select the profile you just created.



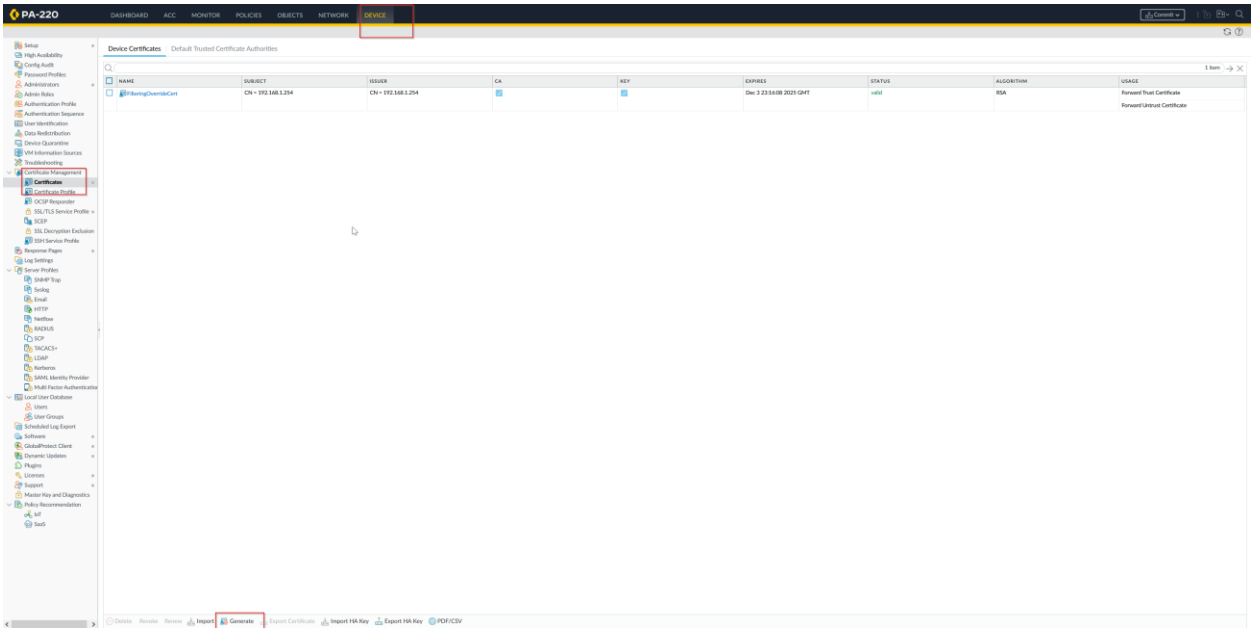
6) Press “Ok” and commit changes. Now, the URL Filtering should work for HTTP websites ONLY. Check that it works like a website like http://guns.com. Note that only some websites allow HTTP traffic, others will automatically redirect to HTTPS, so this test will only work on specific websites.



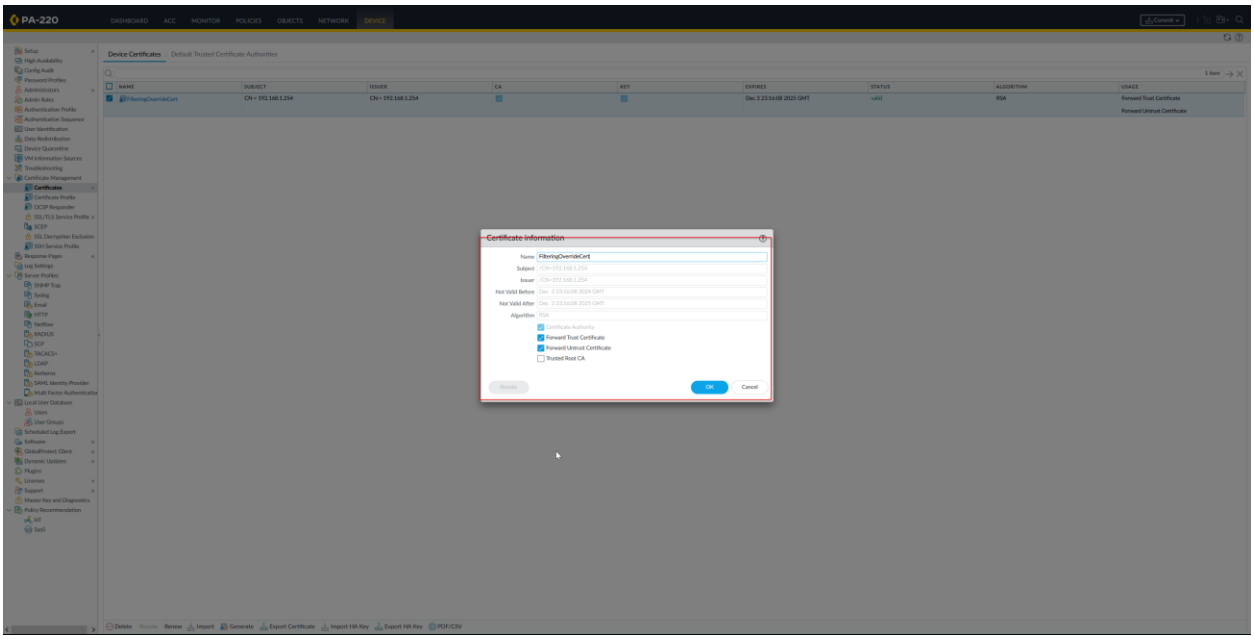
7) Now, we should configure certificates in order to enable HTTPS filtering to work. Console into your firewall and enter this command to enable the firewall to decrypt traffic to insert the block page.

```
# set deviceconfig setting ssl-decrypt url-proxy yes
```

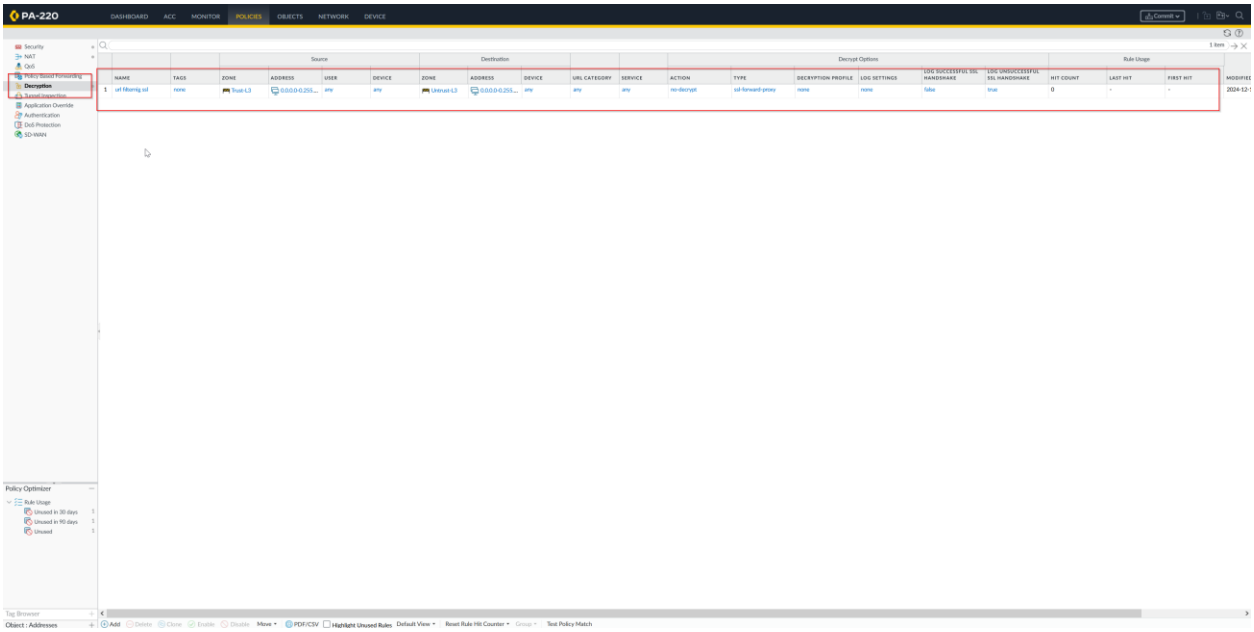
8) Now, we must generate our certificate for the firewall, which will be later imported into our end-devices. Navigate to Device > Certificates > Generate.



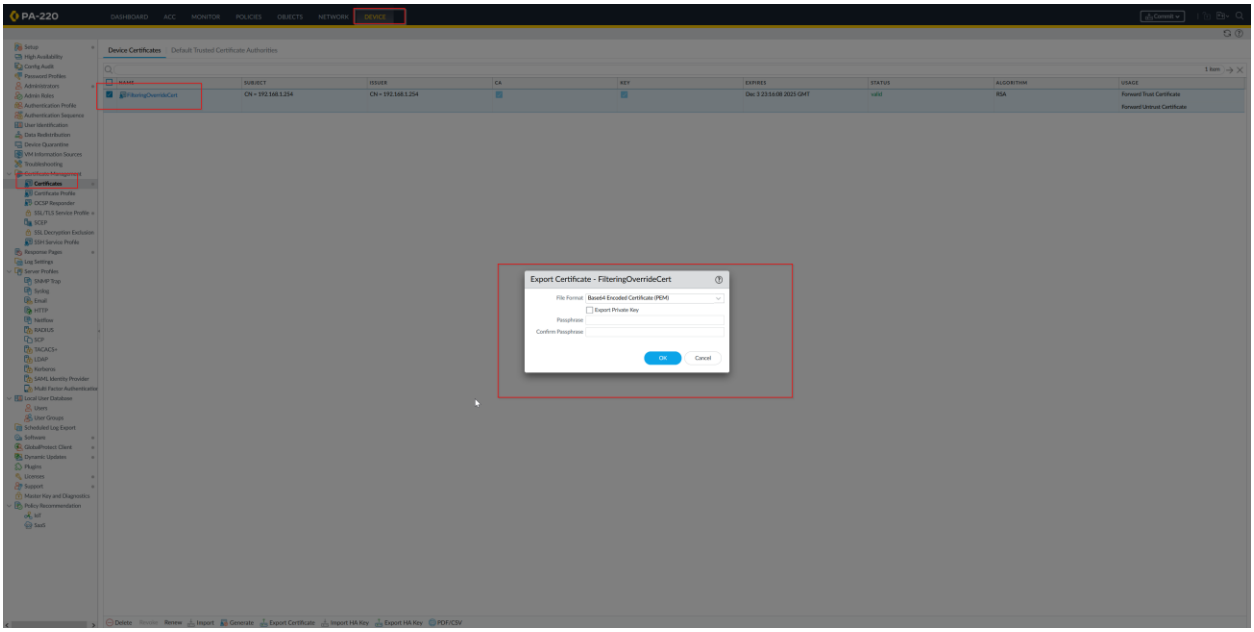
9) Set the following fields. Make sure your default gateway is configured as the correct address.



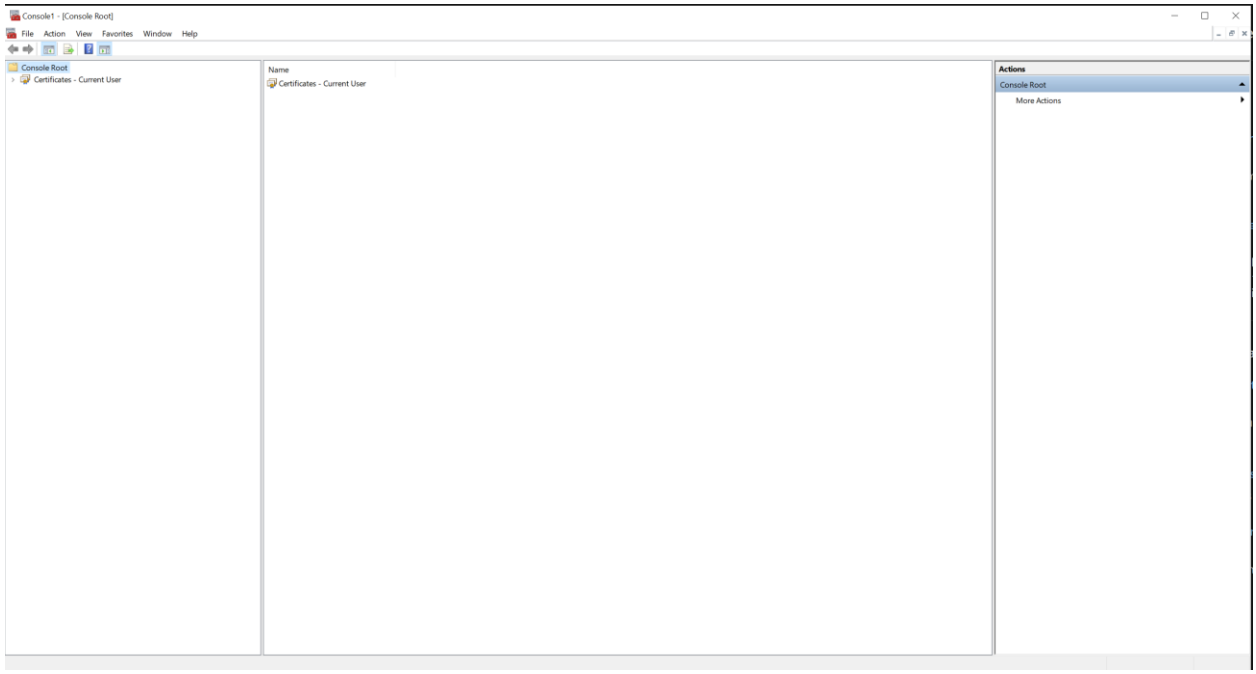
10) Now, go to Policies > Decryption and add a new policy. Make sure it is applied to all inbound and outbound traffic. This tells the firewall to apply decryption to all HTTPS traffic. Set the type as “ssl-decrypt-proxy”. This proxies the HTTPS traffic to our firewall for decryption, where our firewall can then insert the block page.



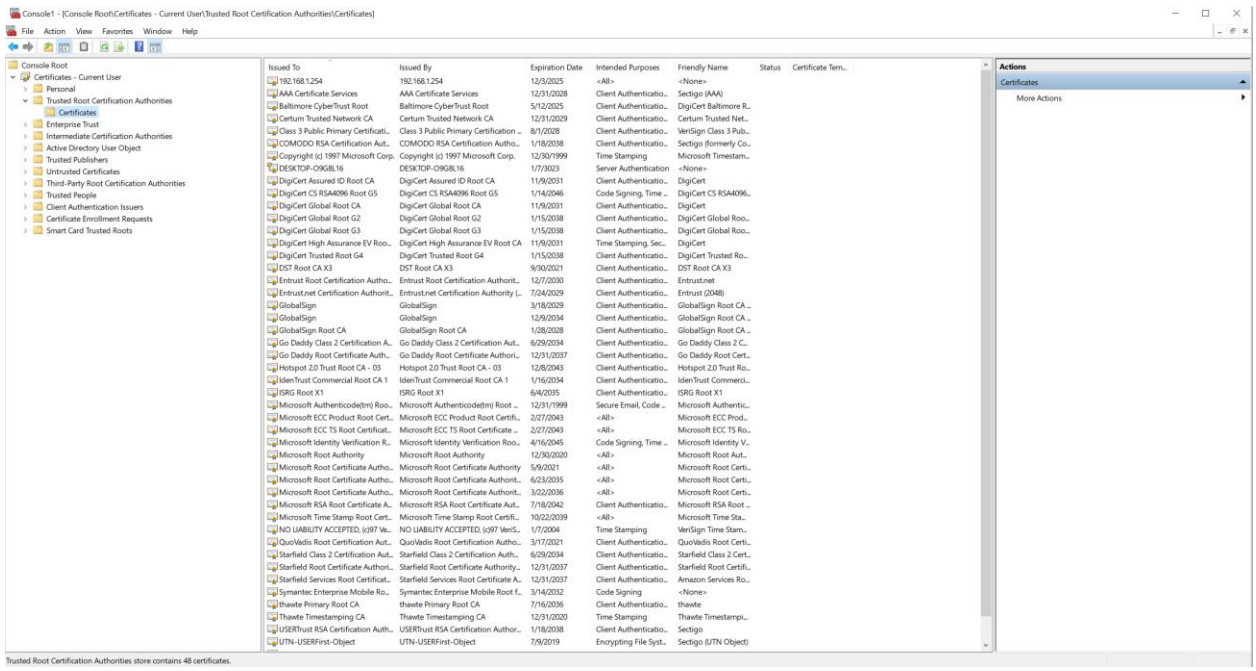
11) Finally, we need to install our Root Certificate on Windows for our end-device. Go back to Device > Certificates > Export Certificate to download the self-signed root certificate.



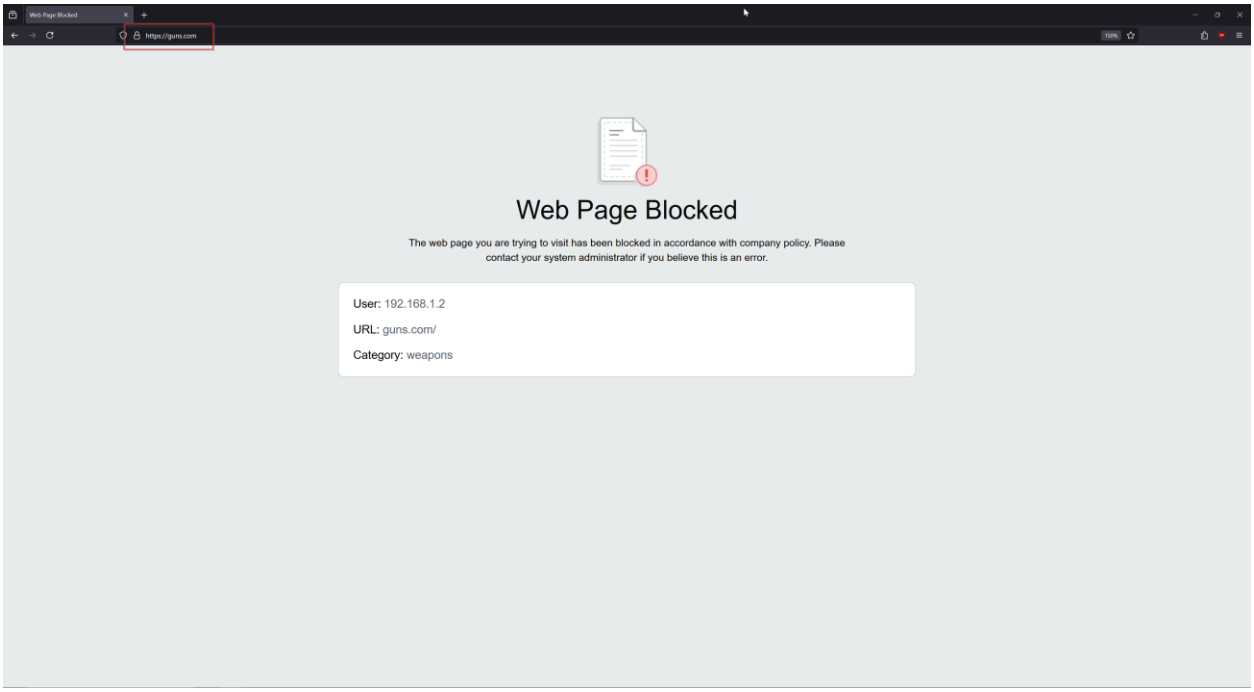
12) Install the root certificate. On Windows, this requires extra permissions. Press “Windows + r” and launch mmc, the Windows management console. Add Certificates to the menu.

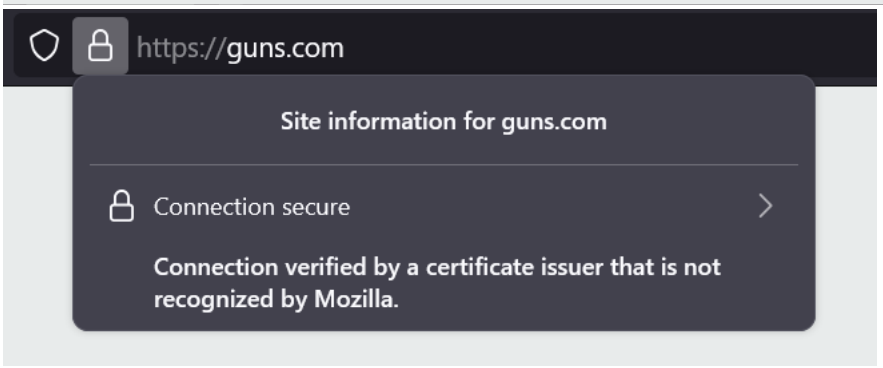


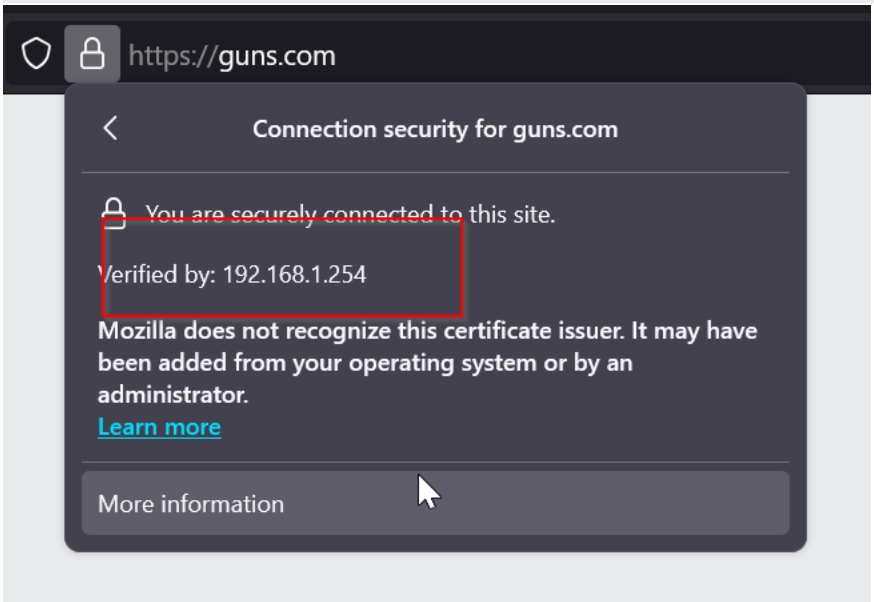
13) Navigate to the Trusted Root Certificate Authorities, and right click to add a new root certificate. Navigate to the download location of the certificate. Add the certificate.



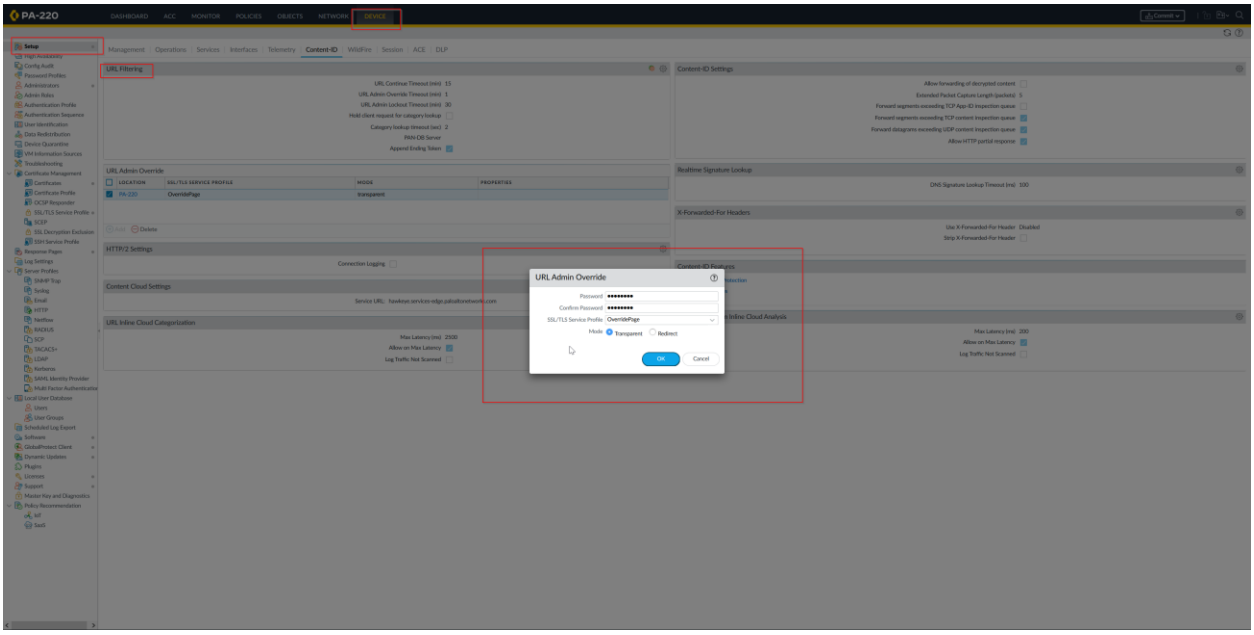
14) The root certificate should now be installed. Verify that HTTPS works with any blocked website.



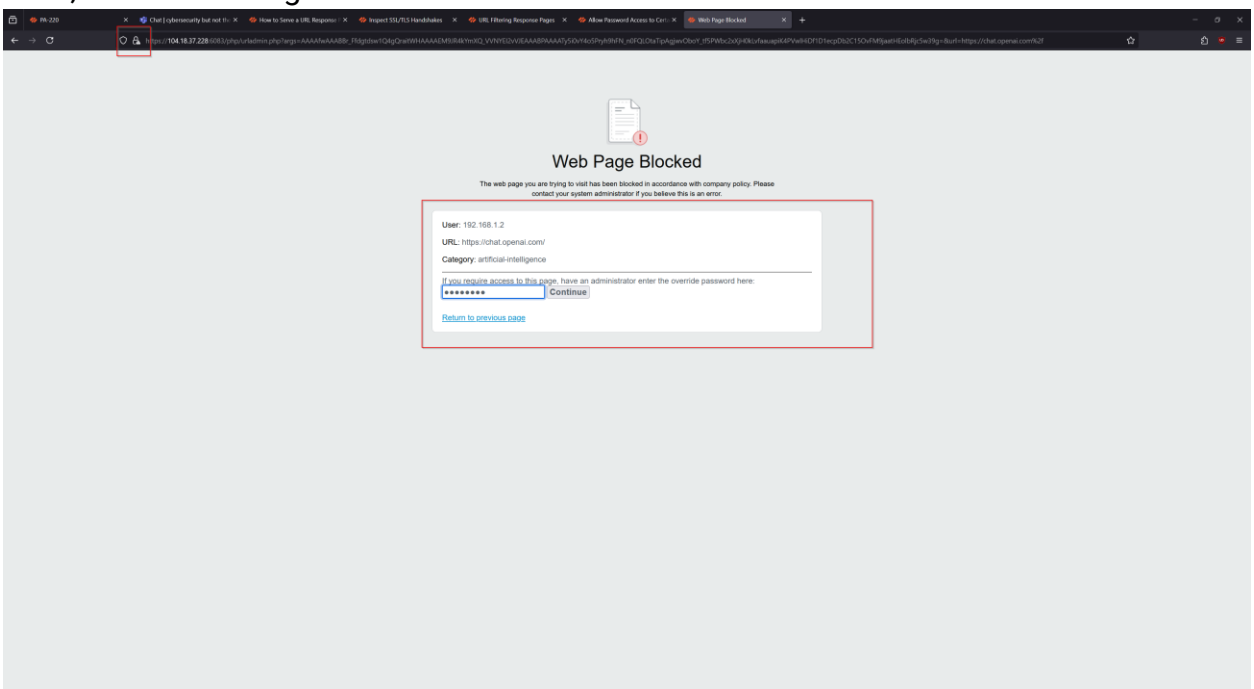




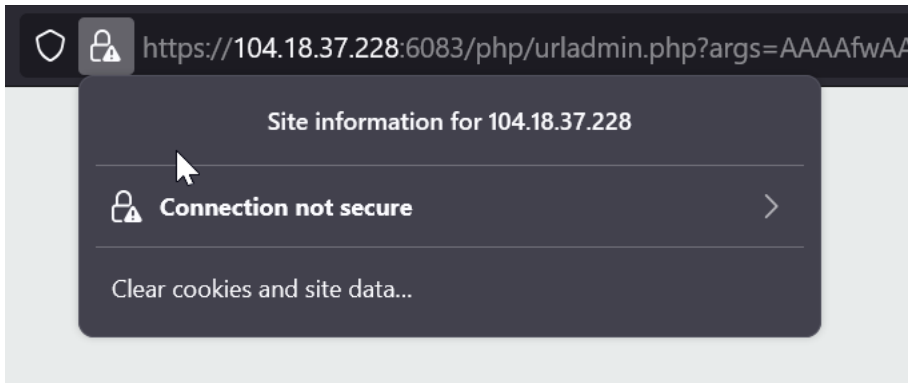
15) We can add the “override” settings now. Go to Device > Setup > URL Filtering and add a new URL Admin Override profile.

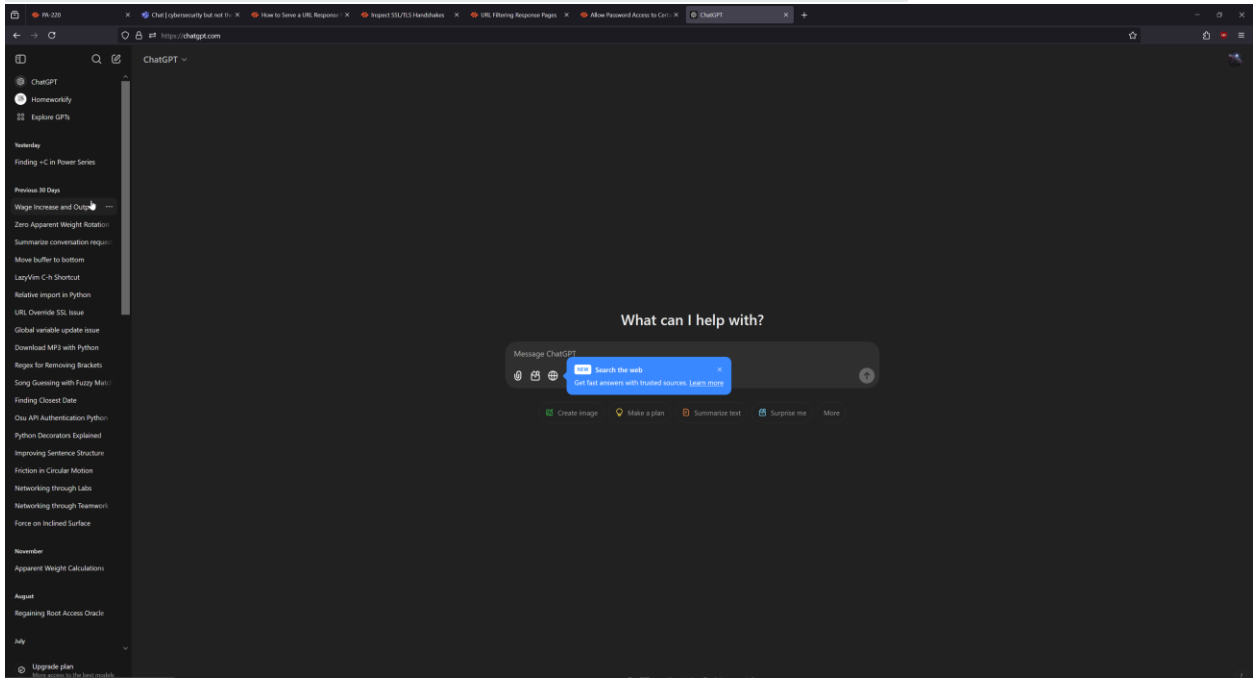


16) Verify that override works with a website that is configured to be override. In this case, artificial intelligence websites are used.



17) Note that the website is shown as “Insecure”, because our certificate is self-signed, however our browser will still allow the connection to go through. URL Filtering is now done!





Problems:

4) **Invalid Palo Alto Device License/No URL Filtering License**

   When we first started the lab, the main issue we had was with the licensing of our device. Since the PA-220 firewalls are handed down each year for the next class of students to configure, licenses may expire if not renewed. As a result, we needed to first obtain/renew our devices URL filtering license BEFORE configuring, as not doing so would result in error messages. Since categorization lists are handled by Palo Alto, licensing is required for our URL Filter to function properly.

5) **SSL Certificate Issues w/HTTPS**

We had a hard time understanding why HTTPS block pages were not working when we first started configuring the lab. Chrome would simply tell us a generic error, and it took us a while to realize that issuing the certificates was the issue. After self-signing and installing the root certificate on our device, the issues were fixed.

Conclusion:

This lab was a great introduction to how URL Filtering functions at a base level and what requirements are necessary for it to work. We were able to learn how HTTPS allows websites to securely transfer data, and certificates are required to enable HTTPS interception, and how block pages functions.

GlobalProtect VPN on Palo Alto PA-220

Ryan Chen

Cybersecurity

Purpose:

This lab simulates a production environment where customers/end-users/employees need to access an internal network from an untrusted, external network. For this purpose, a VPN is used in order to tunnel traffic into the network so that users can access the network from anywhere. Palo Alto provides the GlobalProtect VPN service for this task, and this lab focuses on its configuration and use.

Background Information:

A VPN, or virtual private network, is a commonly used service that provides a multitude of different privacy, security, and convenience functions. At its core, they work by encrypting traffic between an end-user and the server, allowing a secure connection to be established from anywhere. Given that this traffic is encrypted, it is near impossible for an adversary to read it. Even if they do manage to get hold of the traffic by sniffing it off the network, they will be unable to read it.

For this reason, VPN's are commonly used in order to secure traffic from an unsecure network. By establishing this secure connection, attacks like a MITM (Man in the Middle) become less of a threat, since traffic is encrypted. Likewise, people also use VPN's for their ability to "tunnel" traffic. This refers to the ability to establish a connection that emulates the device directly on a local network. This is useful for many different use cases, but commonly required by companies when employees are doing remote work. In order to keep company traffic secure even from an unsecure network, companies can configure a VPN to tunnel employee traffic back into the network. This allows employees to work remotely on company networks, all while keeping traffic secure.

Now that we know the basic use cases of a VPN, we can start to cover how it works at a base level. In order for a VPN to work, end users must connect through an application configured by the VPN provider. This application works by establishing a secure connection through the use of certificates and encryption, allowing end user traffic to be encrypted in transit and decrypted by the VPN at the end. In our case, we generate certificates on the firewall, and then download and install them on our end user devices. This makes our device trust the VPN provider, enabling future communication.

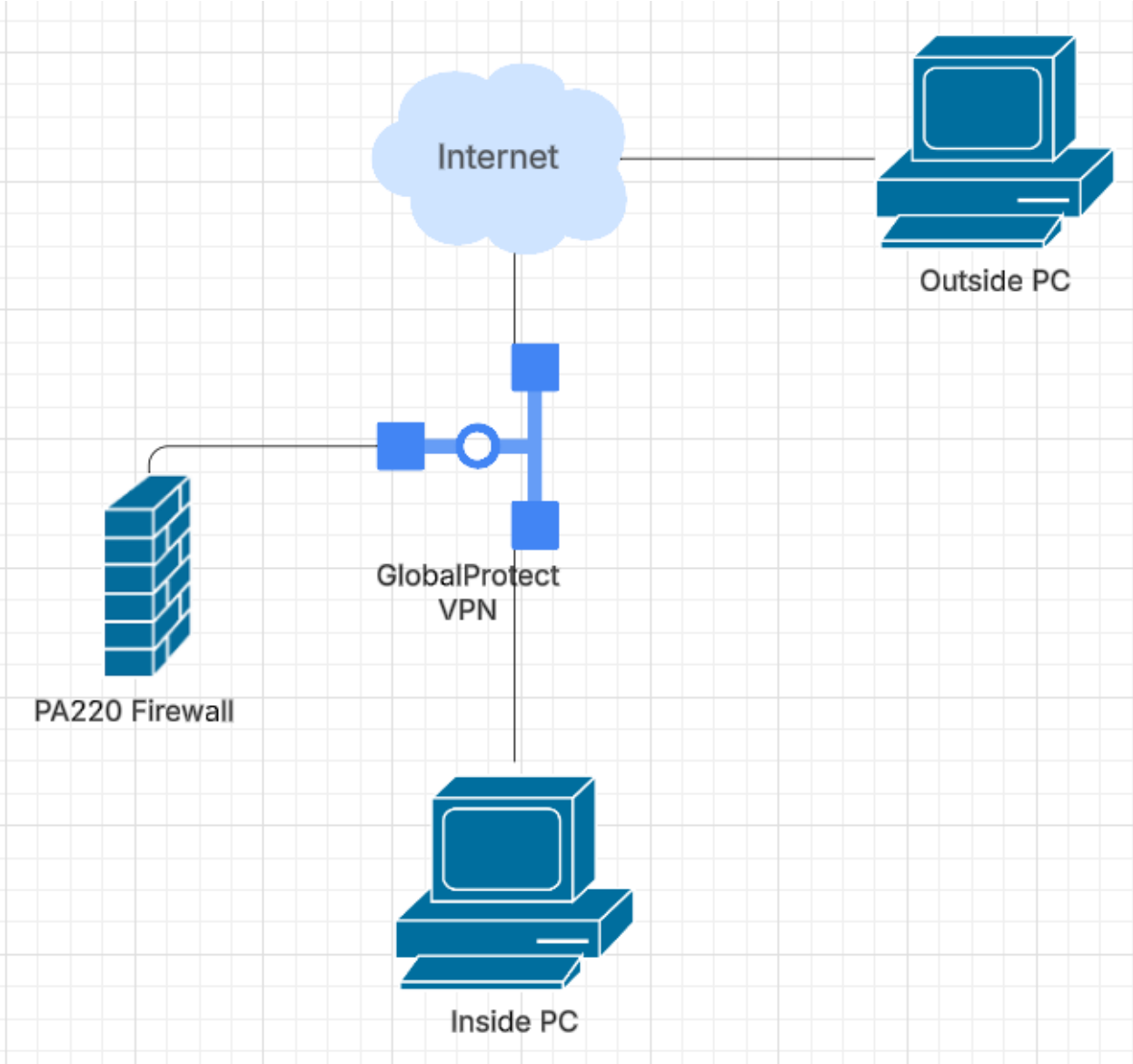
Finally, one of the key concepts of this lab is remote access. For this purpose, we use the RDP (Remote Desktop Protocol) provided by Windows. This protocol allows us to configure one machine inside the network as a server, and the other machine as a end user that will ultimately establish a connection to the server to demonstrate that the VPN is working. RDP gives us the ability to remotely view another Windows desktop, a common use case for companies that want to enable employees to do remote work.

Lab Summary:

As previously mentioned, the first step to our lab is to generate and install certificates on both our firewall and our end user devices. This is crucial to allowing the VPN to ultimately establish a connection between server and client through an insecure intermediate network. Next, we need to create a tunnel interface that all VPN traffic will go through. This is important, as the tunnel needs to allow traffic to flow from an insecure region (outward facing interface) into our private network. We will use this later when configuring our client.

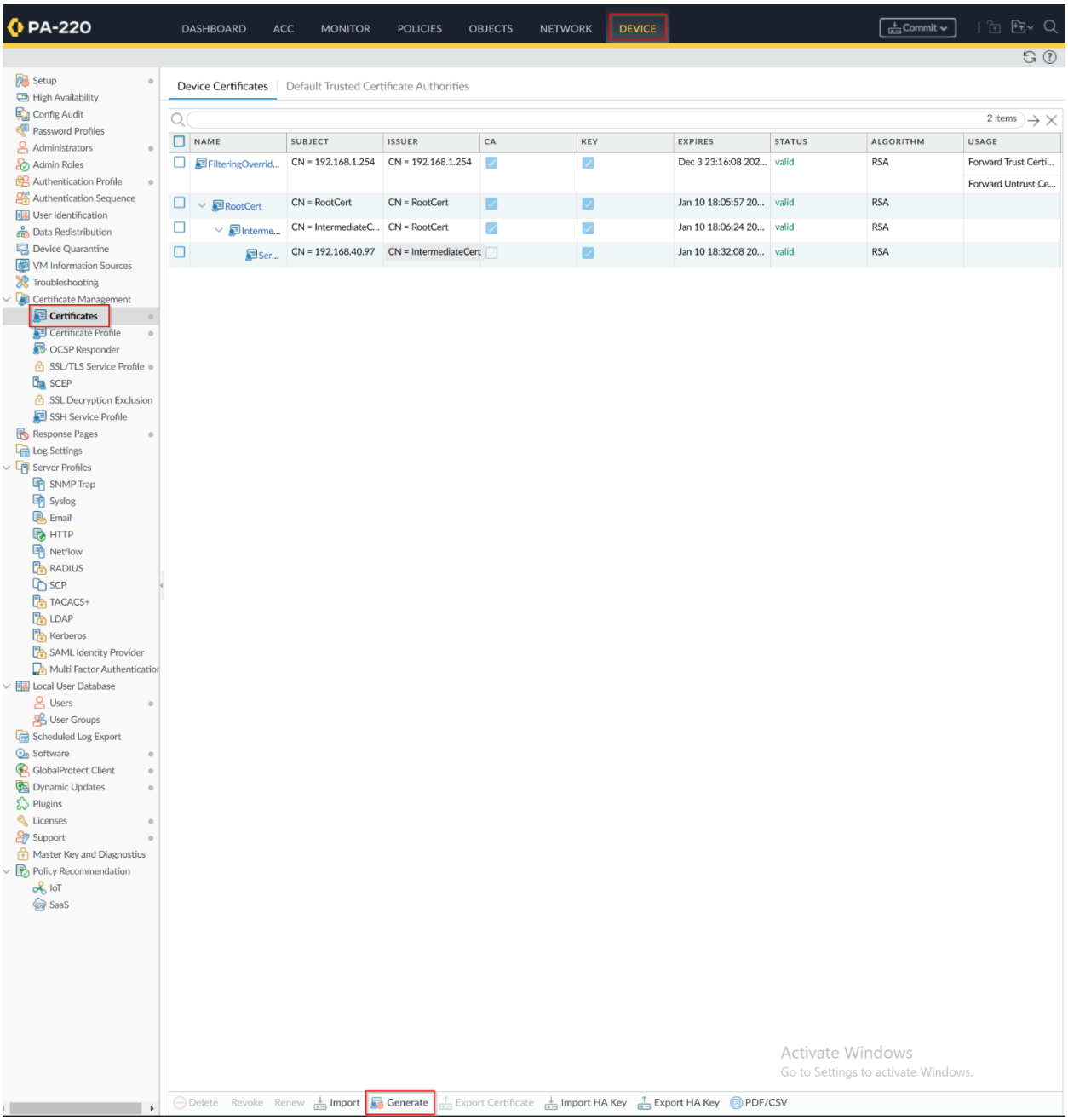
Finally, the actual VPN server has to be configured. To simplify deployment, we use Palo Alto's GlobalProtect service, making a VPN much easier to create. After configuring this, we should be done with the VPN side of our lab. After allowing RDP on one of our machines, we should be able to connect using the VPN into the local network, from where we can RDP to prove that the VPN works.

Network Diagram:

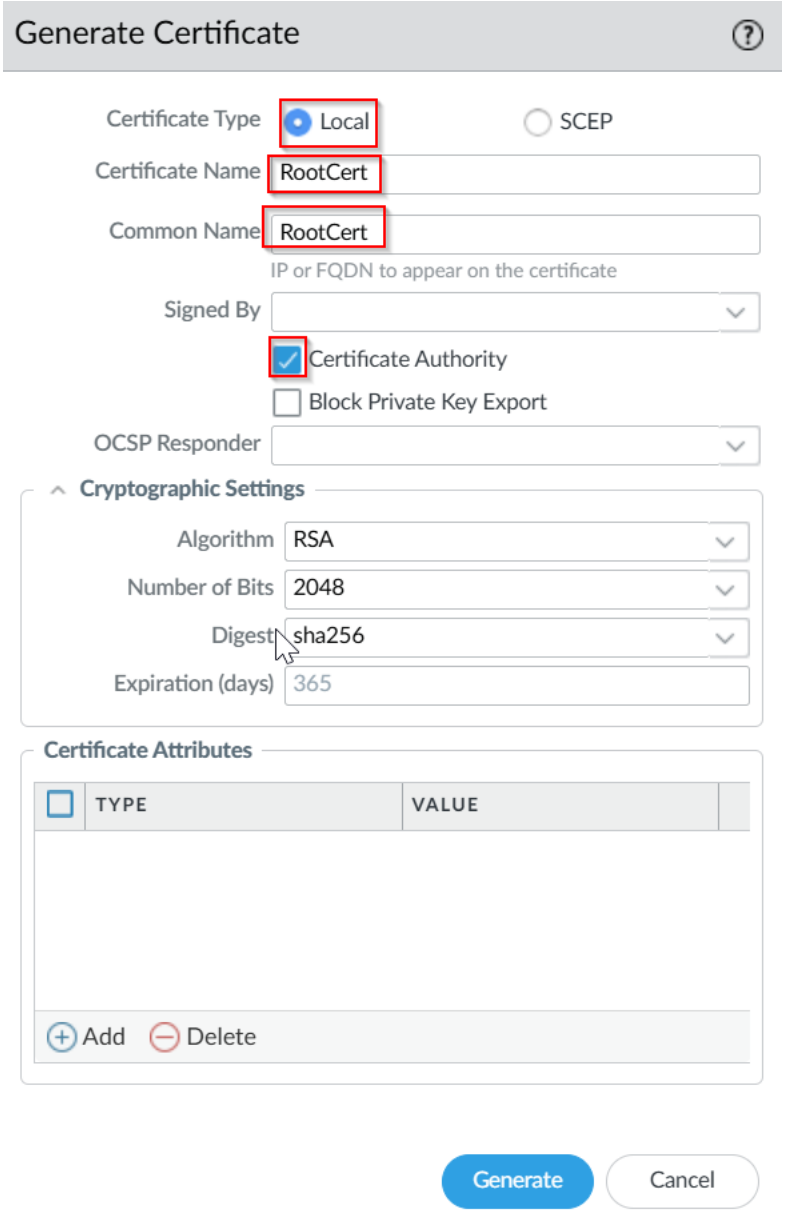


Lab Commands:

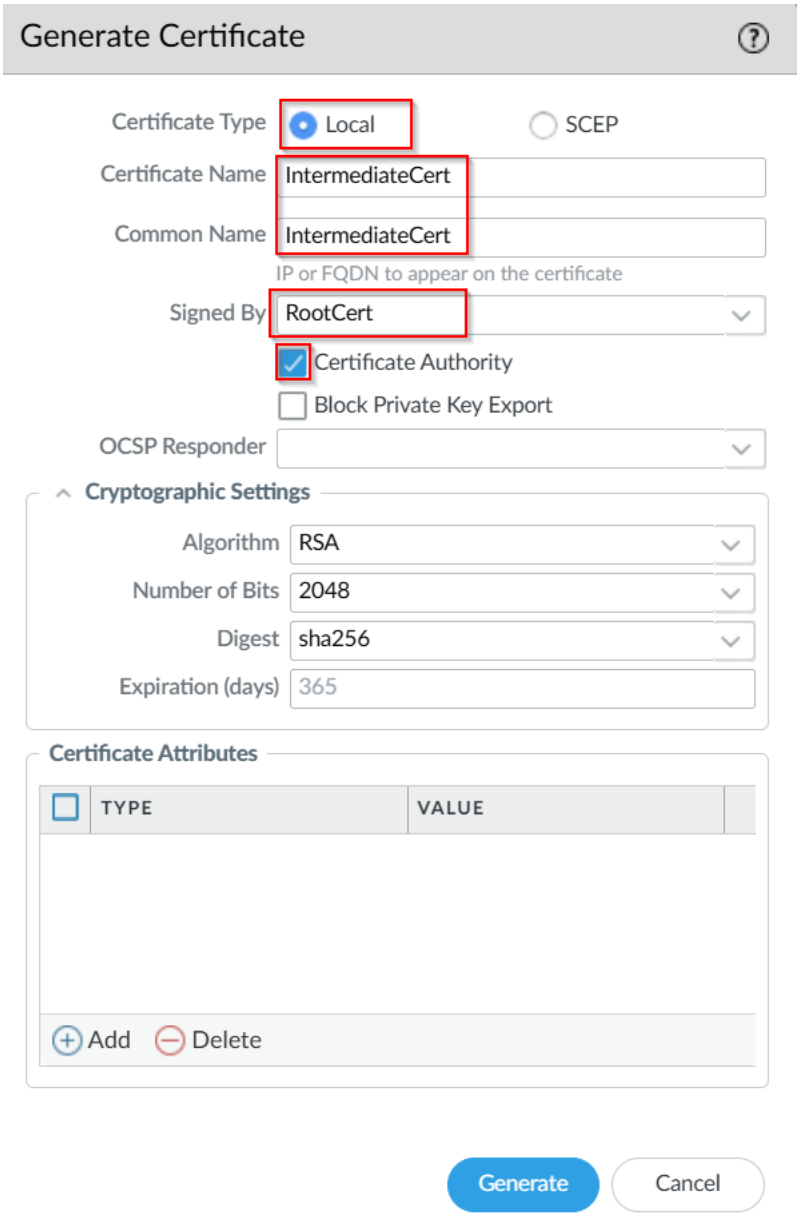
1) To begin, we need to generate certificates that will be used in order to prove that our server is trustworthy to the client. These include a root and intermediate certificate that the client end user machines will later download and import. Navigate to Certificates > Generate.



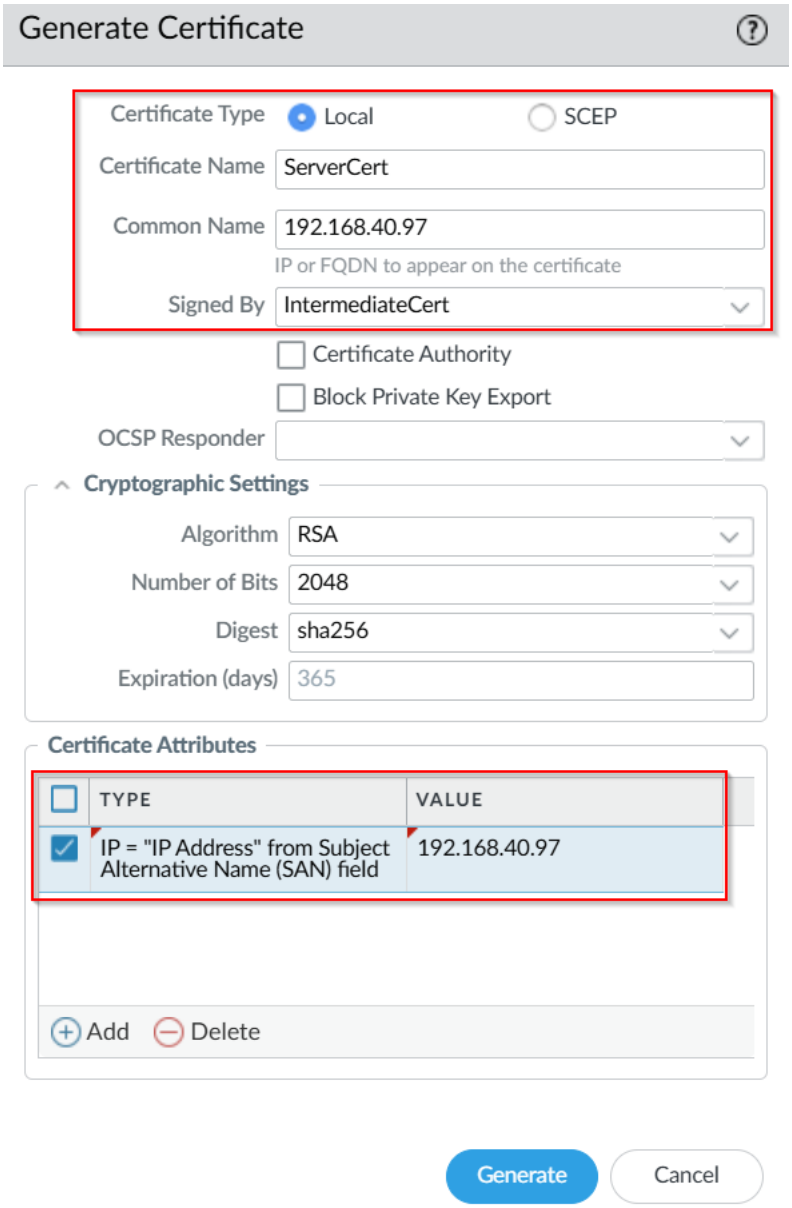
2) First, create the root certificate. Note that it must be a certificate authority.



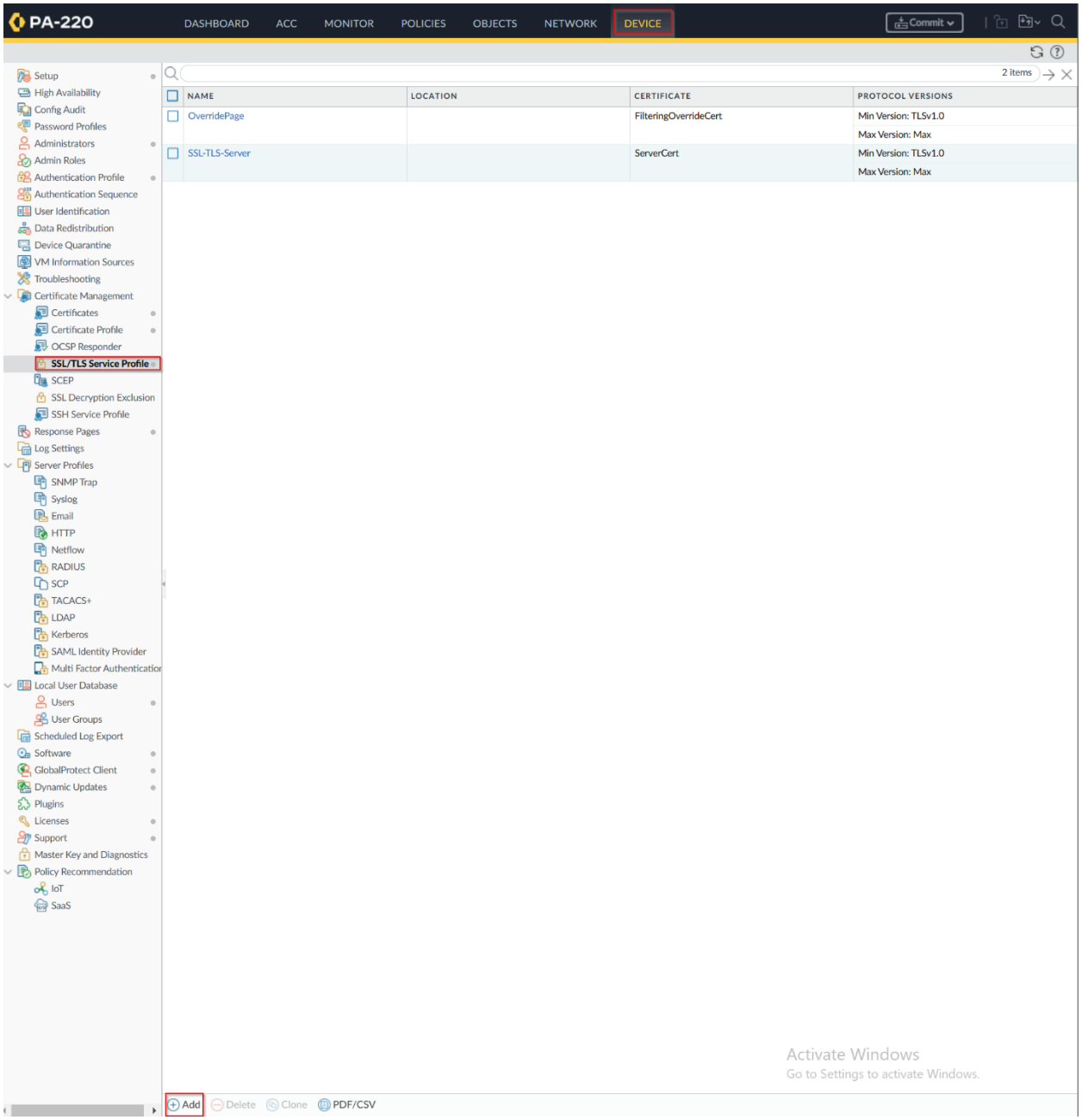
3) Next, generate an Intermediate certification. This should be signed by the root certificate and also be a certificate authority, which will be necessary later for end devices.



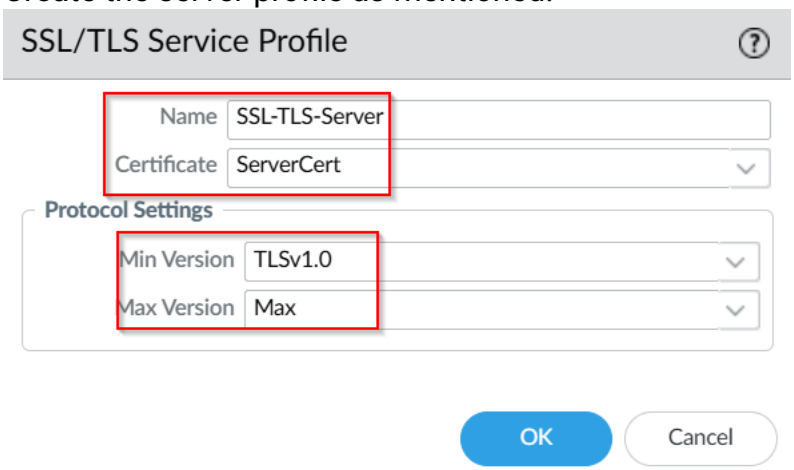
4) Finally, create a server certificate. Note that this will not be installed on end users' devices and thus does not need to be a certificate authority. Make sure to point this certificate to the corresponding IP address of your outward facing interface.



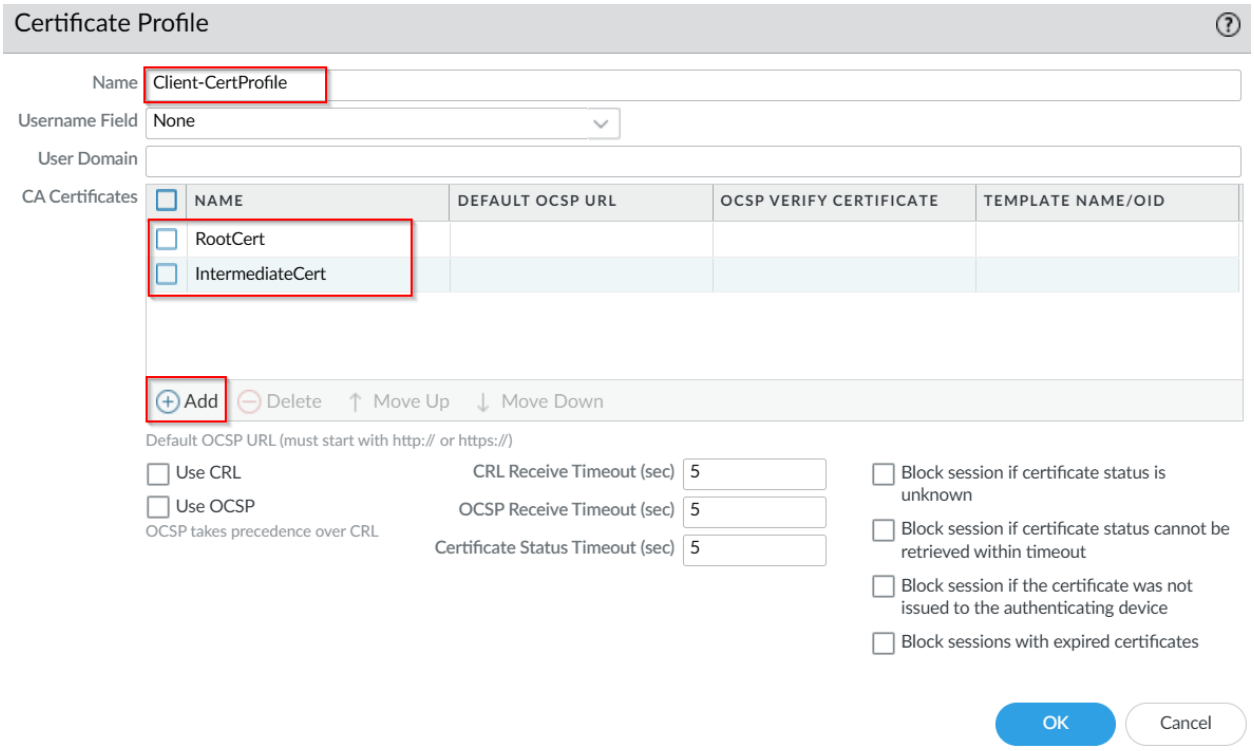
5) Next, we need to navigate to SSL/TLS Service Profiles and create a new profile for our VPN to encrypt traffic. This was previously done in URL filtering, but needs to be done again.


PA-220
DASHBOARD
ACC
MONITOR
POLICIES
OBJECTS
NETWORK
DEVICE
Commit
Setup
High Availability
Config Audit
Password Profiles
Administrators
Admin Roles
Authentication Profile
Authentication Sequence
User Identification
Data Redistribution
Device Quarantine
VM Information Sources
Troubleshooting
Certificate Management
Certificates
Certificate Profile
OCSP Responder
SSL/TLS Service Profile
SCEP
SSL Decryption Exclusion
SSH Service Profile
Response Pages
Log Settings
Server Profiles
SNMP Trap
Syslog
Email
HTTP
Netflow
RADIUS
SCP
TACACS+
LDAP
Kerberos
SAML Identity Provider
Multi Factor Authenticatior
Local User Database
Users
User Groups
Scheduled Log Export
Software
GlobalProtect Client
Dynamic Updates
Plugins
Licenses
Support
Master Key and Diagnostics
Policy Recommendation
IoT
SaaS
2 items
NAME
LOCATION
CERTIFICATE
PROTOCOL VERSIONS
OverridePage
FilteringOverrideCert
Min Version: TLSv1.0
Max Version: Max
SSL-TLS-Server
ServerCert
Min Version: TLSv1.0
Max Version: Max
Activate Windows
Go to Settings to activate Windows.
Add
Delete
Clone
PDF/CSV

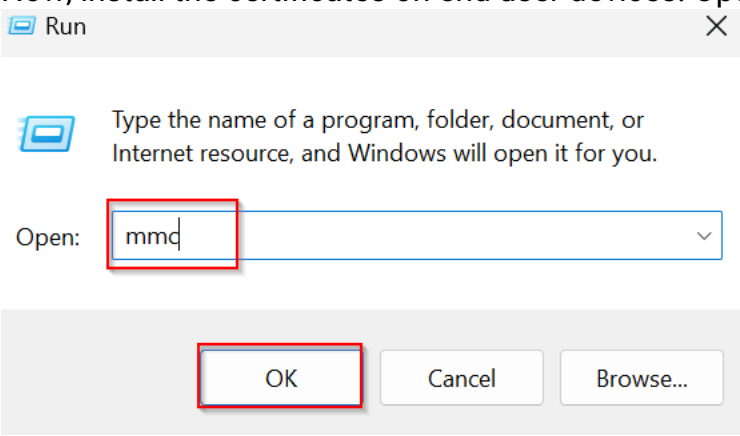
6) Create the server profile as mentioned.



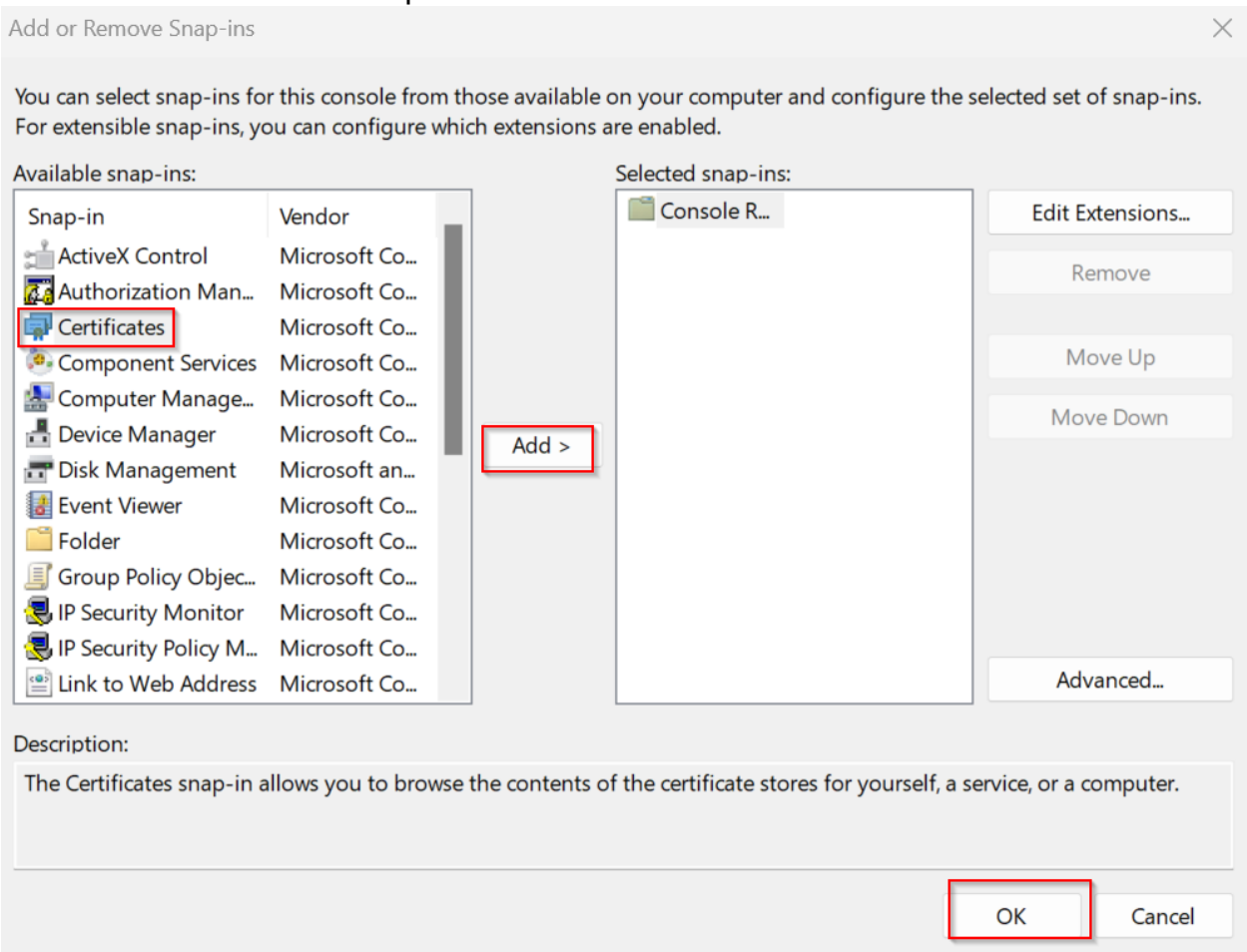
7) The server profile should include the root certificate and the intermediate certificate.

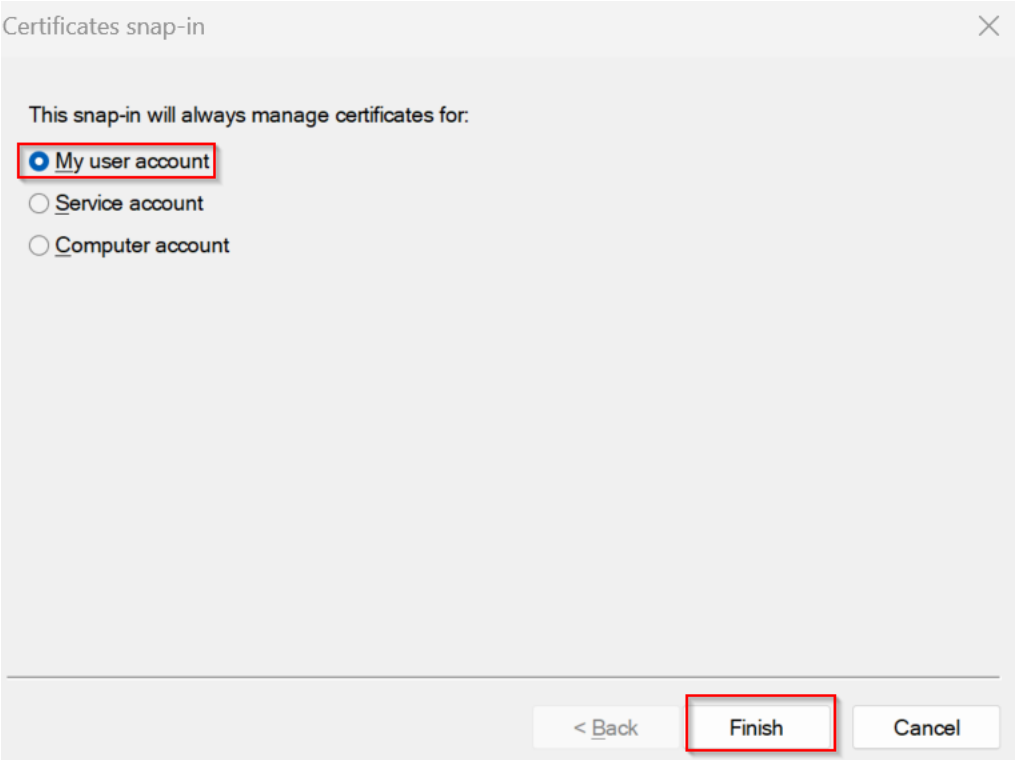


8) Now, install the certificates on end user devices. Open Windows MMC.

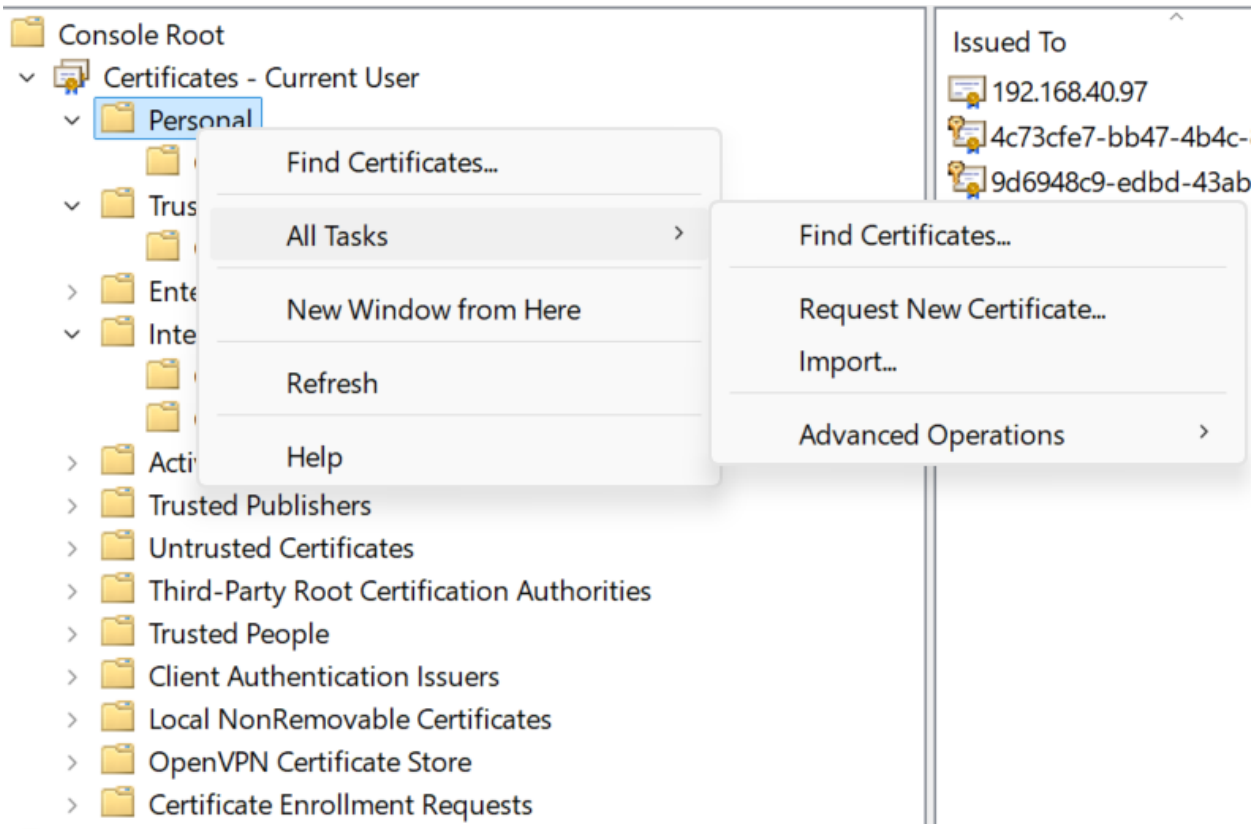


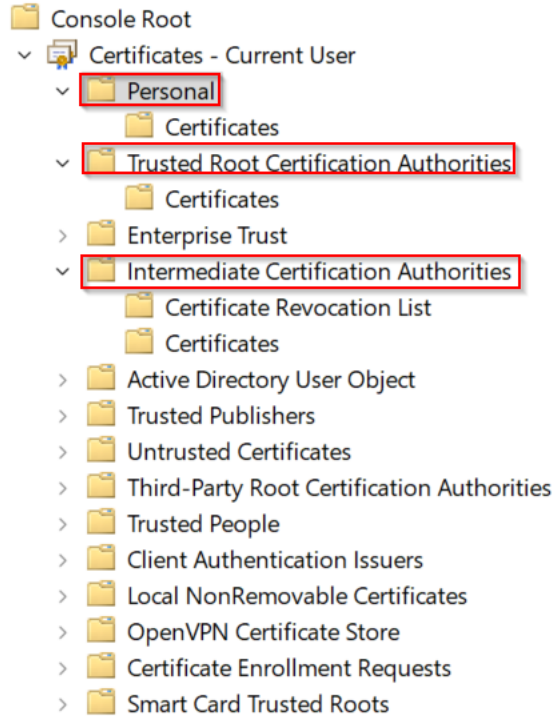
9) Add the certificates as a snap in and set it to be local.



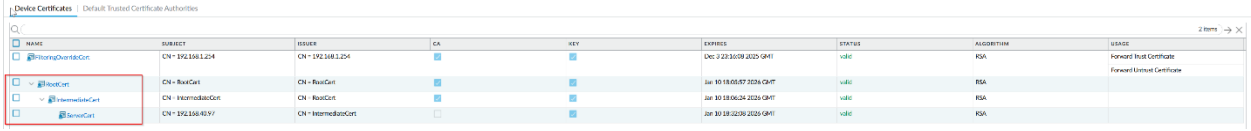

Certificates snap-in
This snap-in will always manage certificates for:
My user account
Service account
Computer account
< Back
Finish
Cancel

10) Install the certificates as Trusted Root certificates.

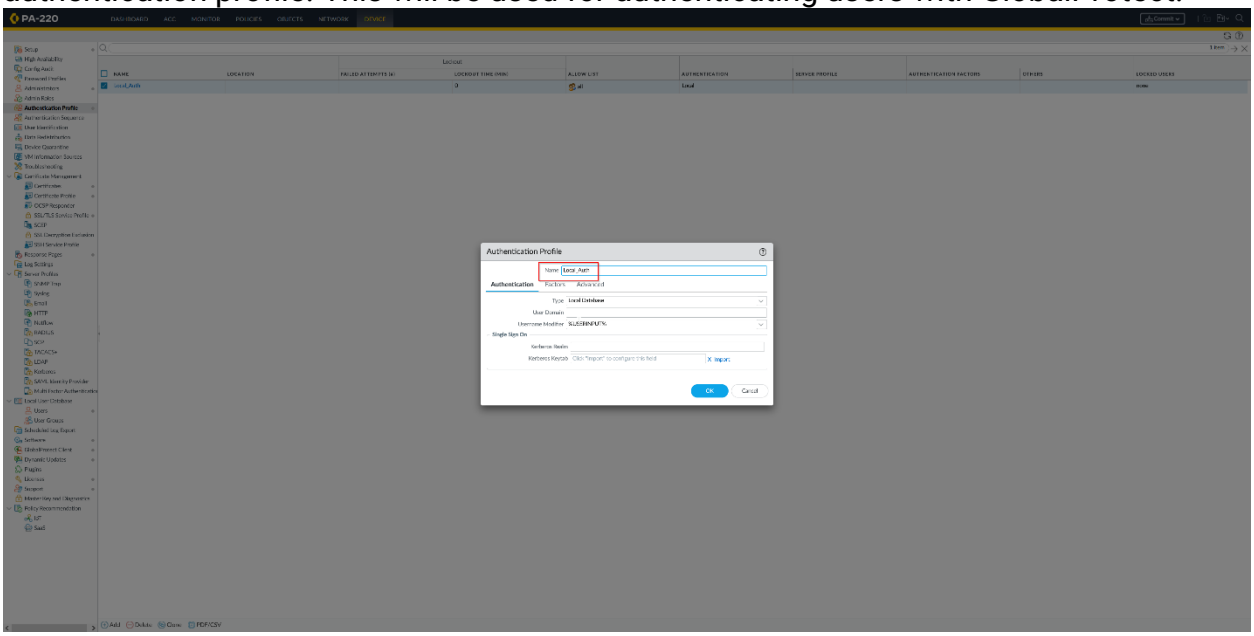




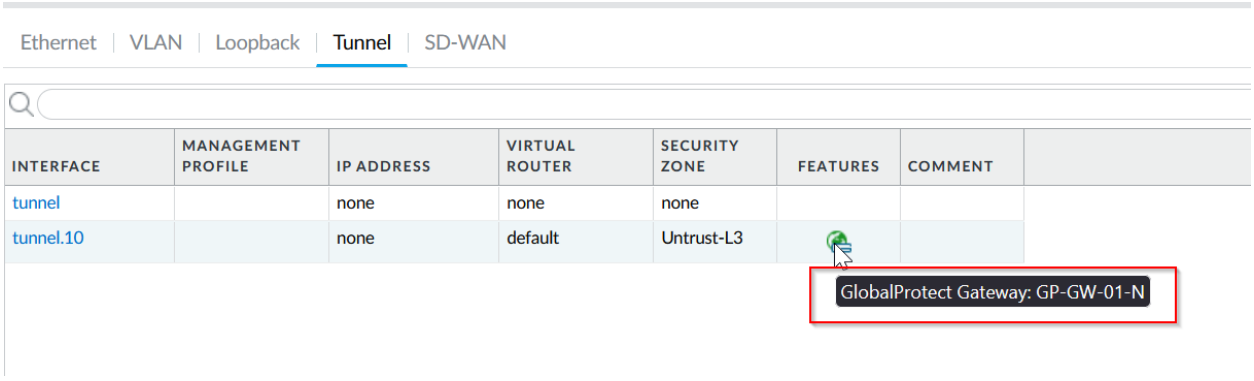
11) Now, we can continue with setting up the actual GlobalProtect. Verify that the Device Certificates are correctly created as so. They should be in a hierarchy.



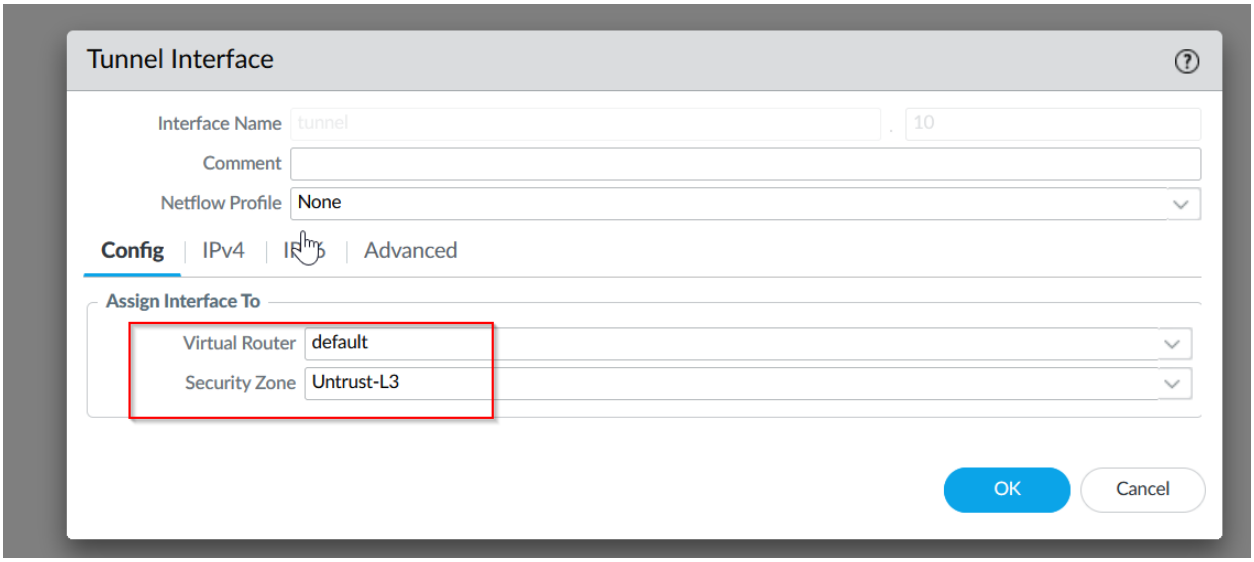
12) Now, navigate to Device > Authentication Profile. Create a new “Local_Auth” authentication profile. This will be used for authenticating users with GlobalProtect.



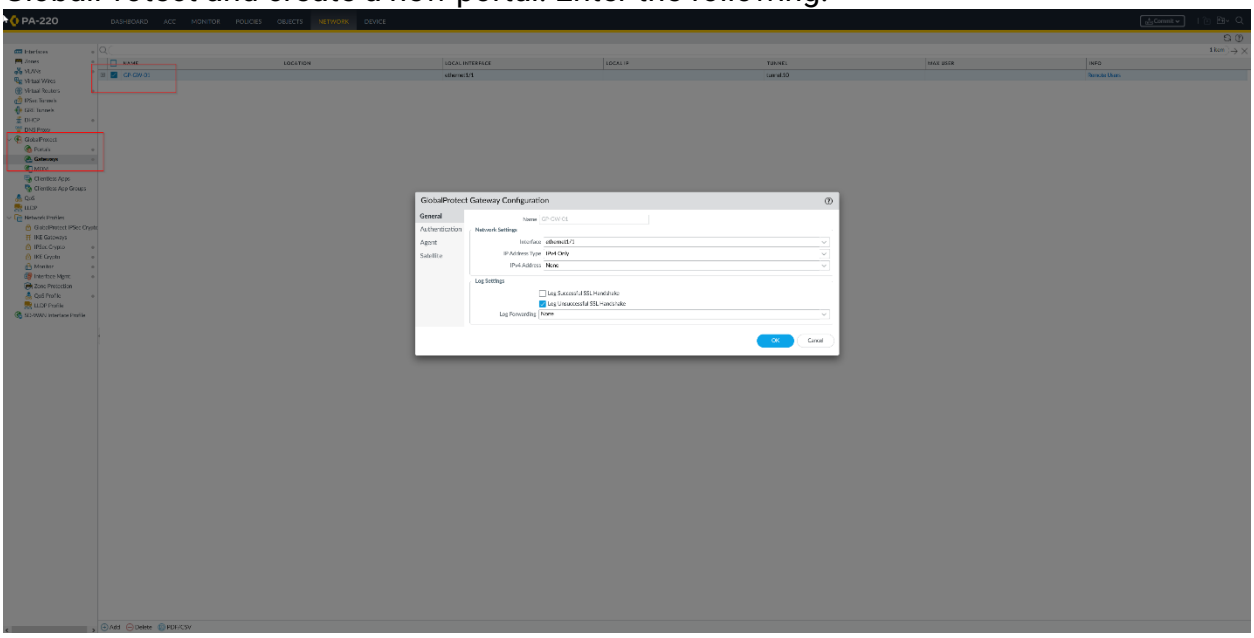
13) Next, create a tunnel for the GlobalProtect traffic. This interface will be used to “tunnel” the traffic from the insecure outside network into the inside network.

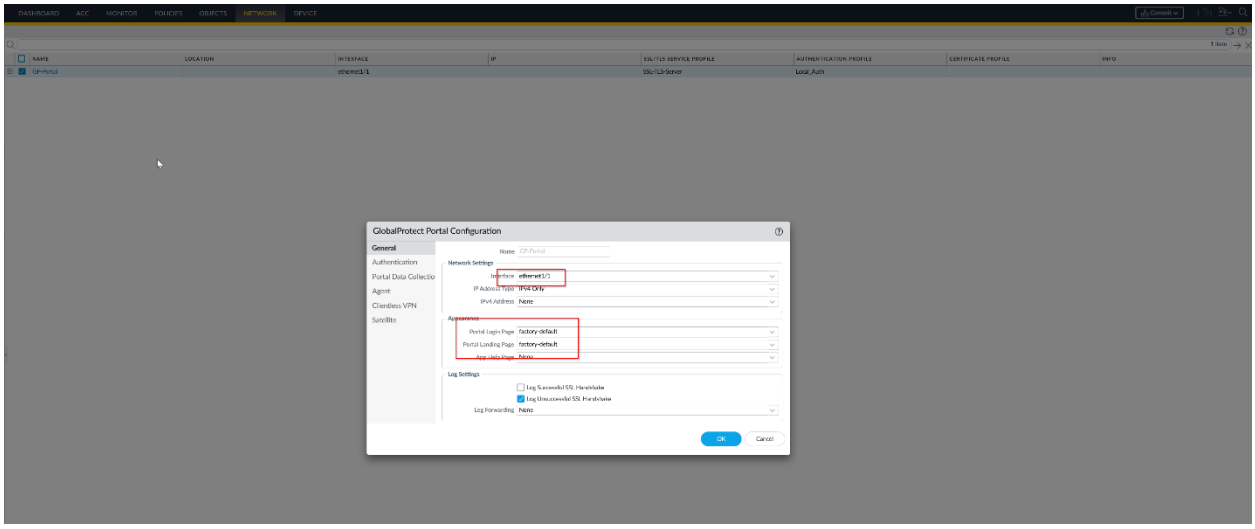


14) The tunnel should be set as a Untrust-L3 zone, since it is taking in traffic from the outside.

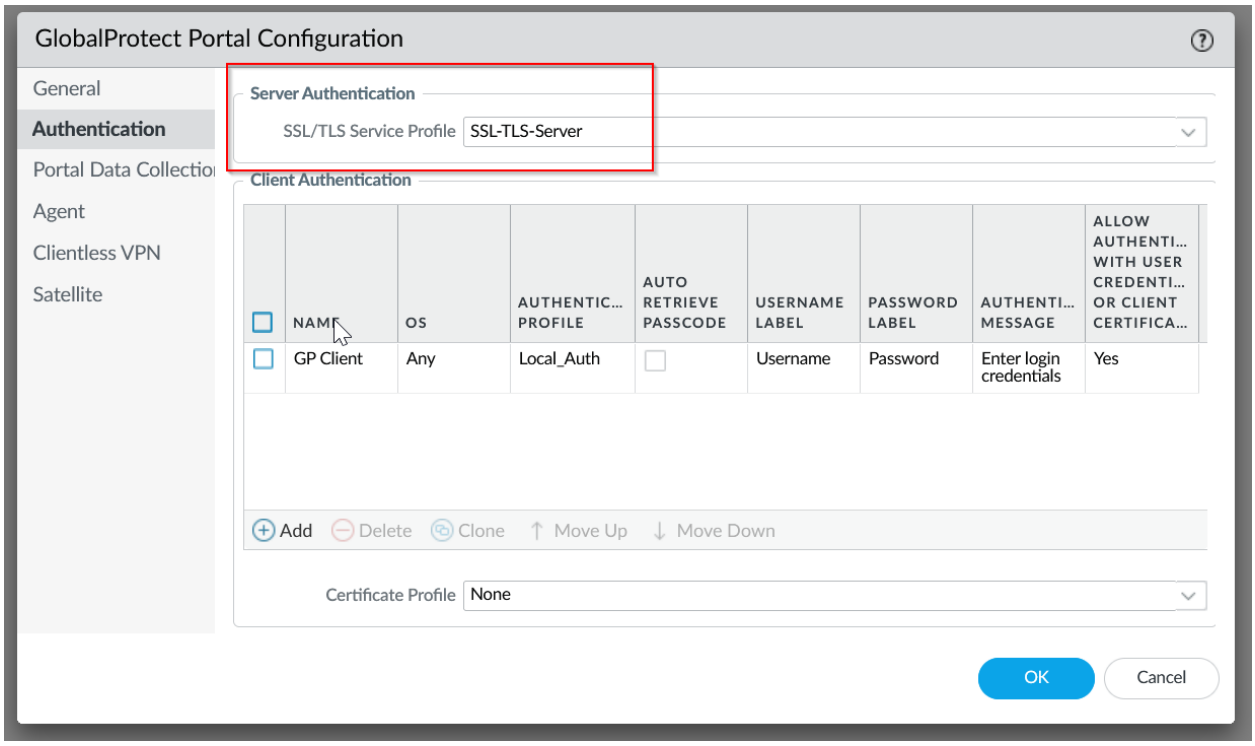


15) Finally, we can create our GlobalProtect portal. Navigate to Network > GlobalProtect and create a new portal. Enter the following:

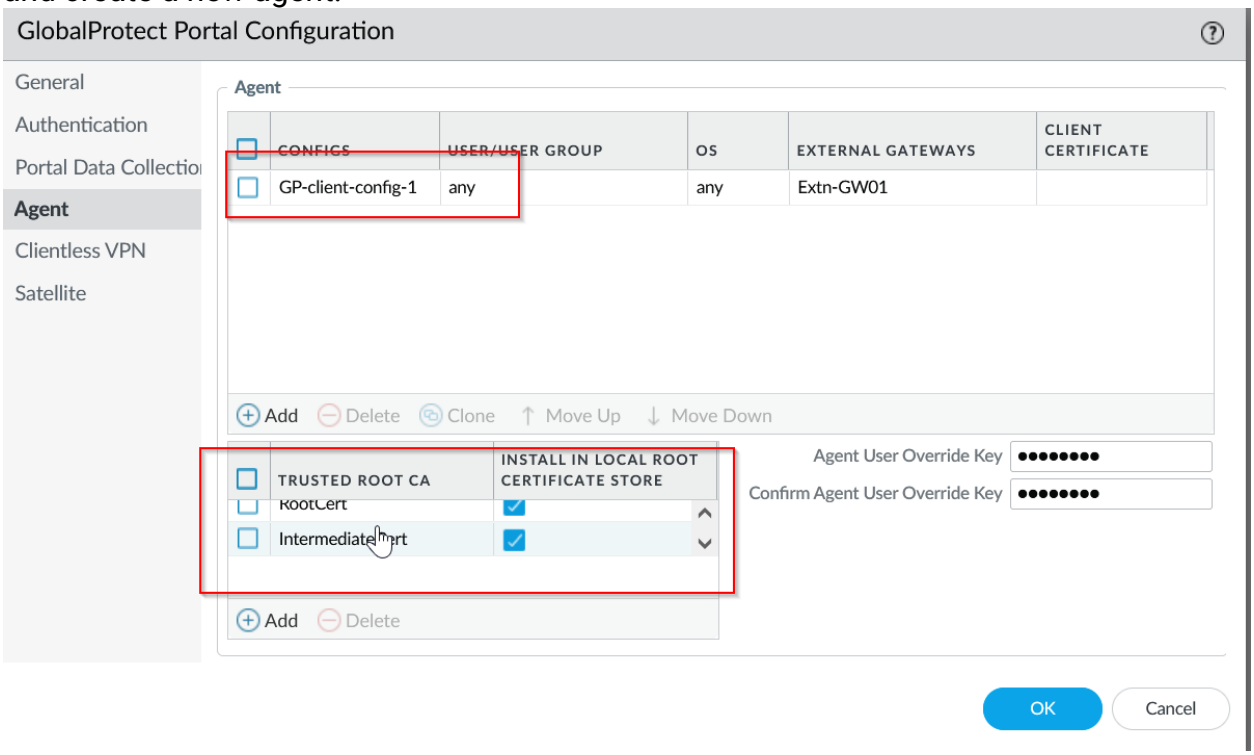




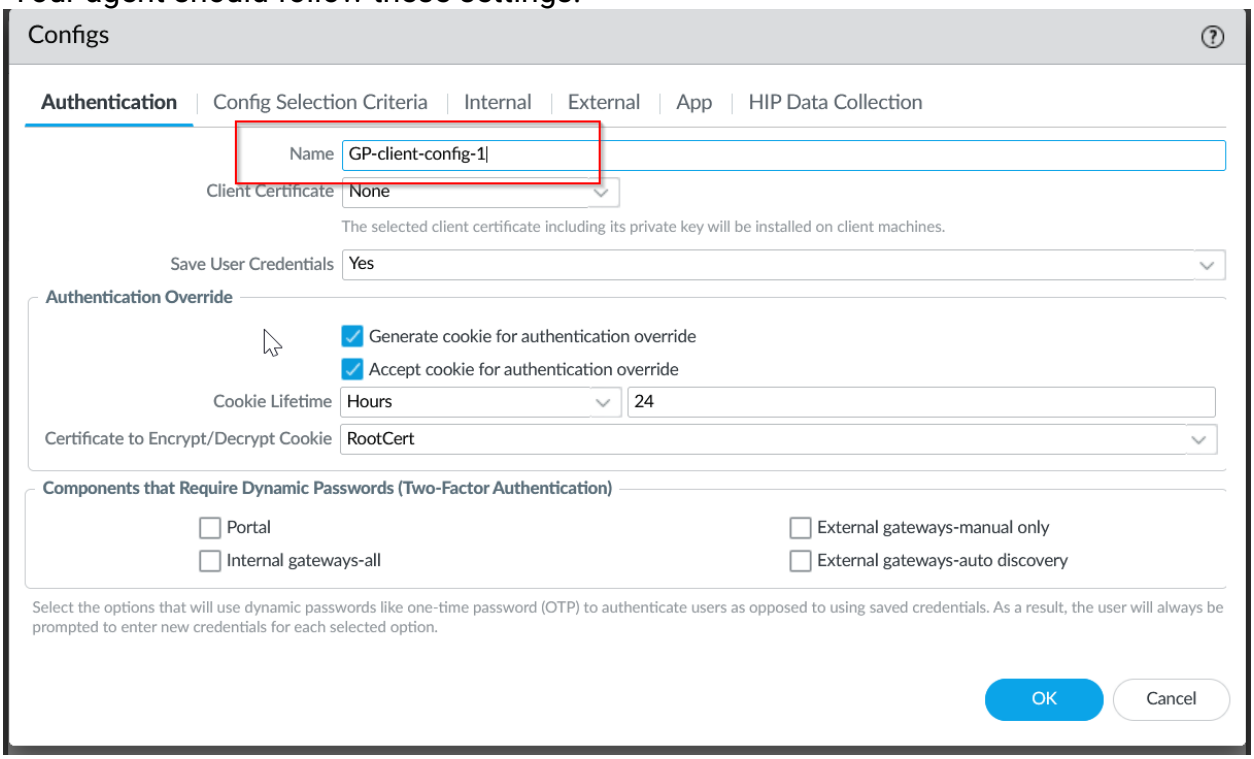
16) Select the SSL/TLS profile that was created in the previous steps for the authentication section.



17) For the agent section, add your root and intermediate certifications created earlier, and create a new agent.

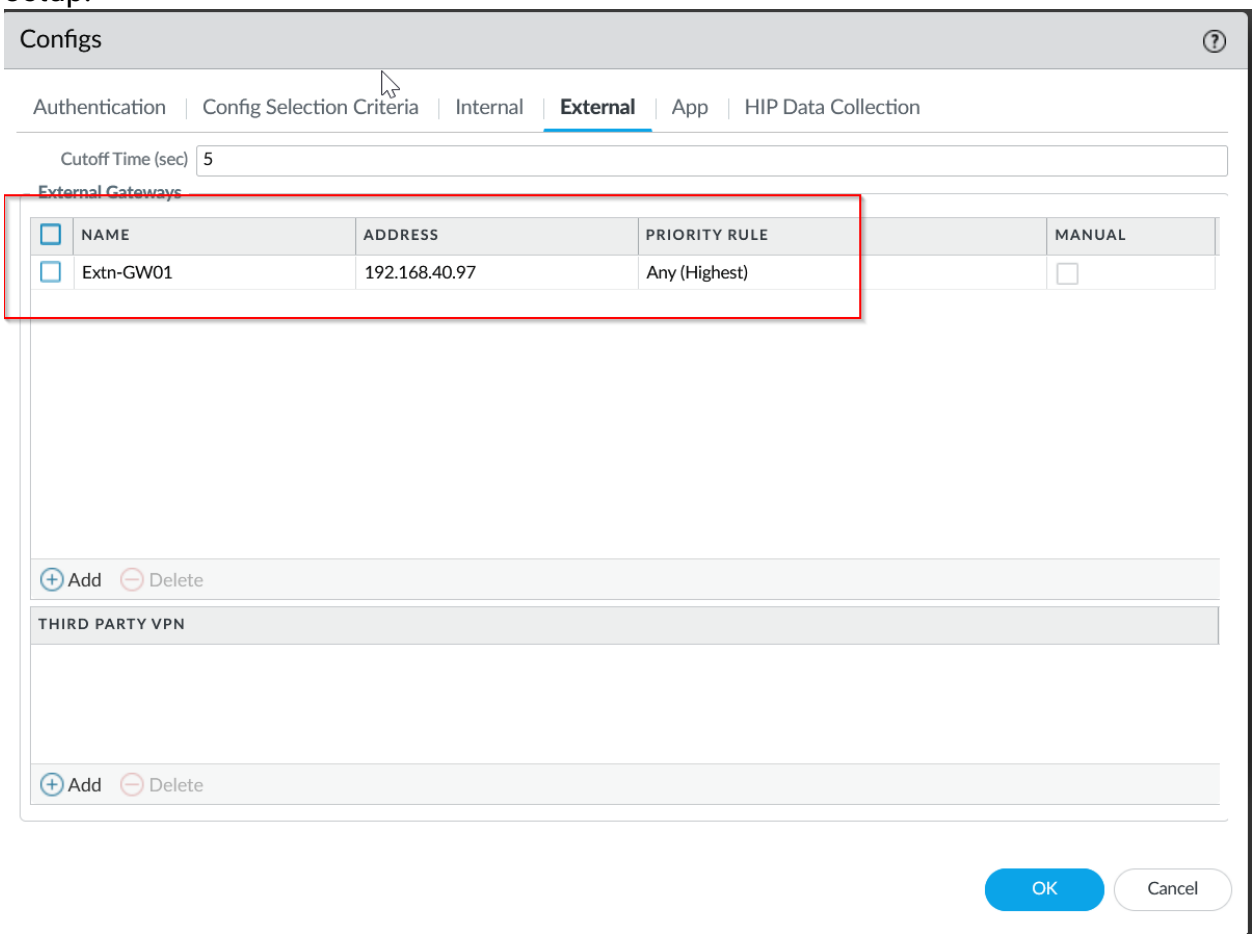


18) Your agent should follow these settings:

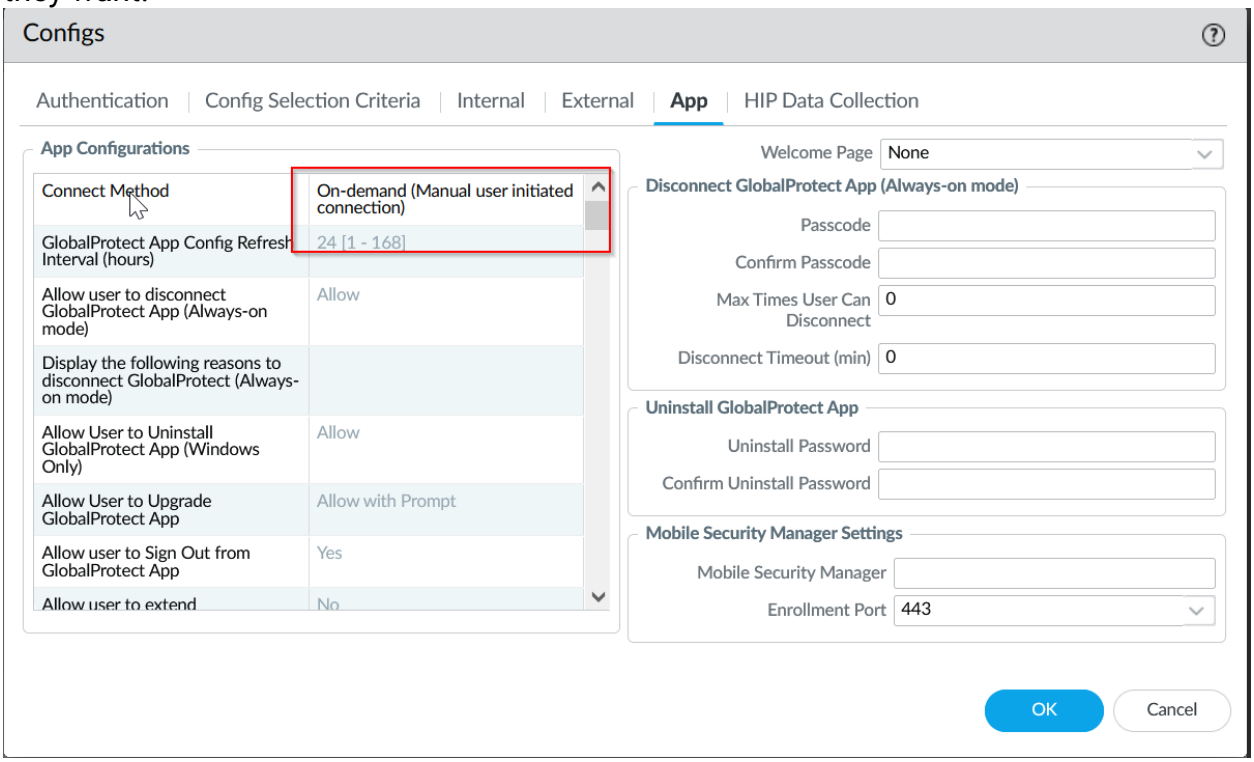


19) Make sure to set the external gateway. This is important! It must be the IP address of your outward facing address, otherwise end users will be unable to reach your

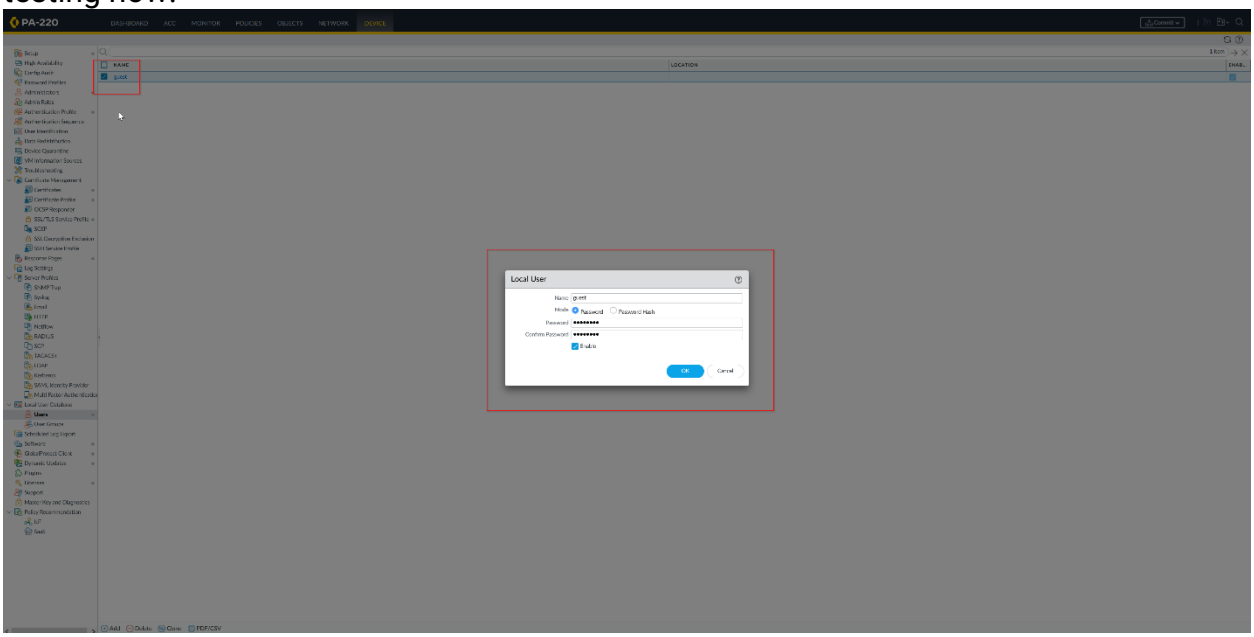
setup.


Configs
Authentication
Config Selection Criteria
Internal
External
App
HIP Data Collection
Cutoff Time (sec)
5
External Gateways
NAME
ADDRESS
PRIORITY RULE
MANUAL
Extn-GW01
192.168.40.97
Any (Highest)
Add
Delete
THIRD PARTY VPN
Add
Delete
OK
Cancel

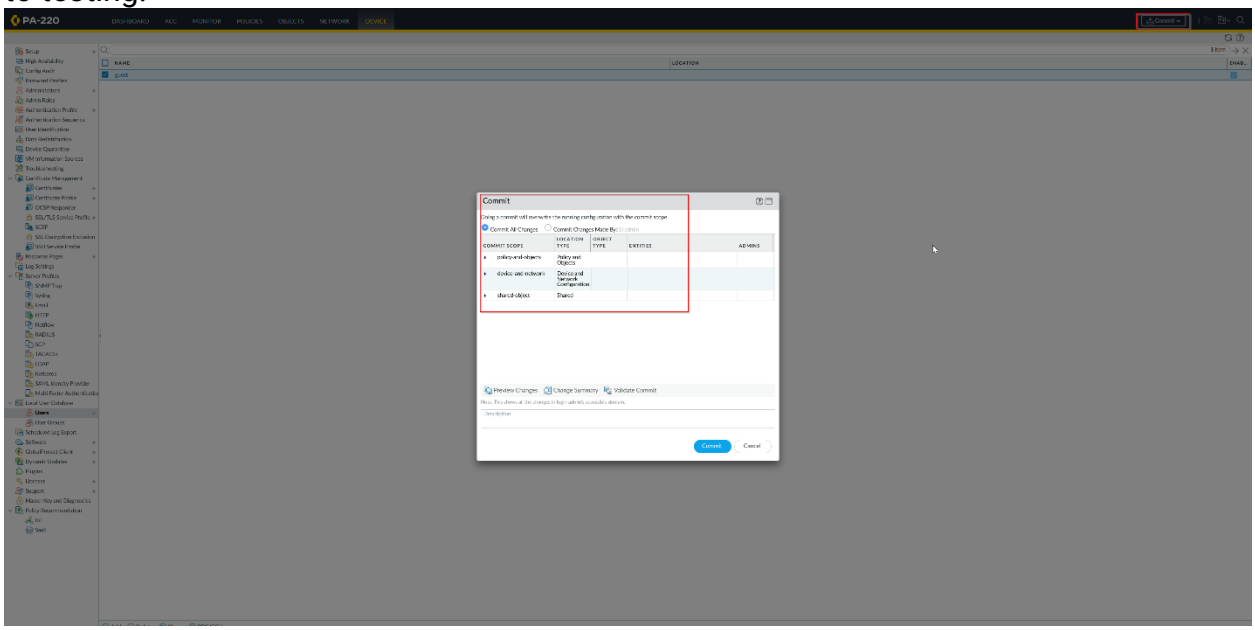
20) The connection type should be On Demand, allowing users to connect whenever they want.



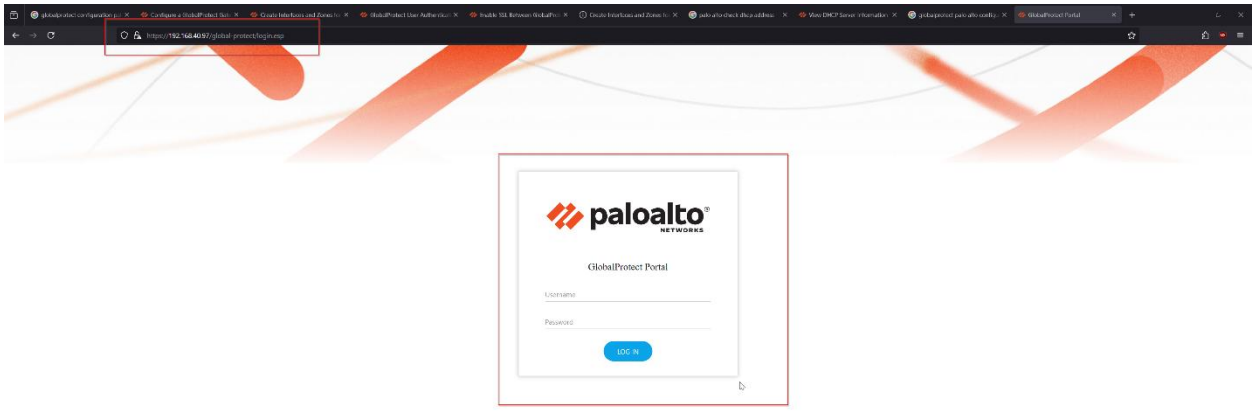
21) Create it, and navigate to Device > Users. We can finally create our local user for testing now.



22) After we finish this, we are done with configuration. Commit changes and proceed to testing.

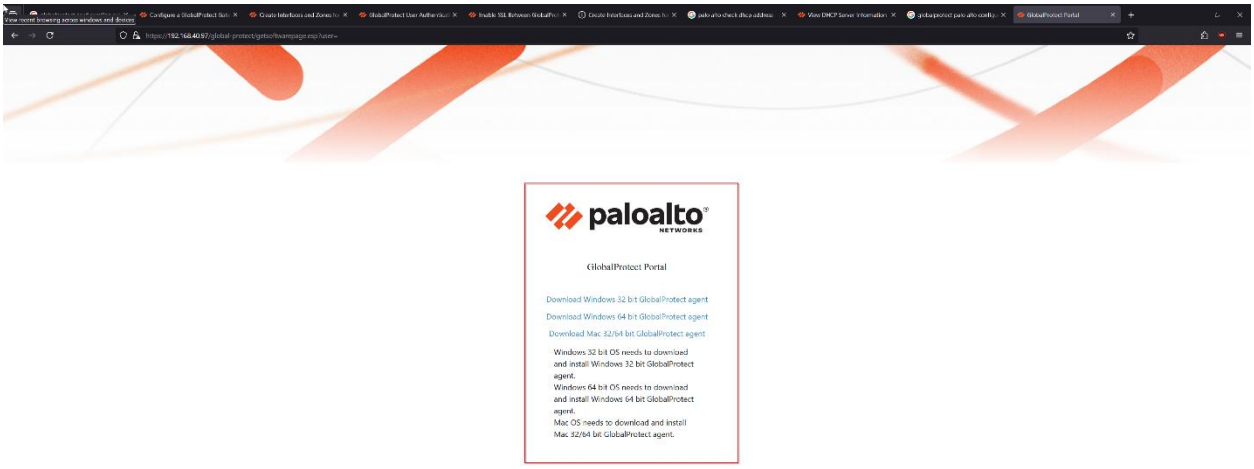


23) Navigate to the outward facing IP address. It should show the GlobalProtect portal. This is where we will download our VPN client on our end devices.

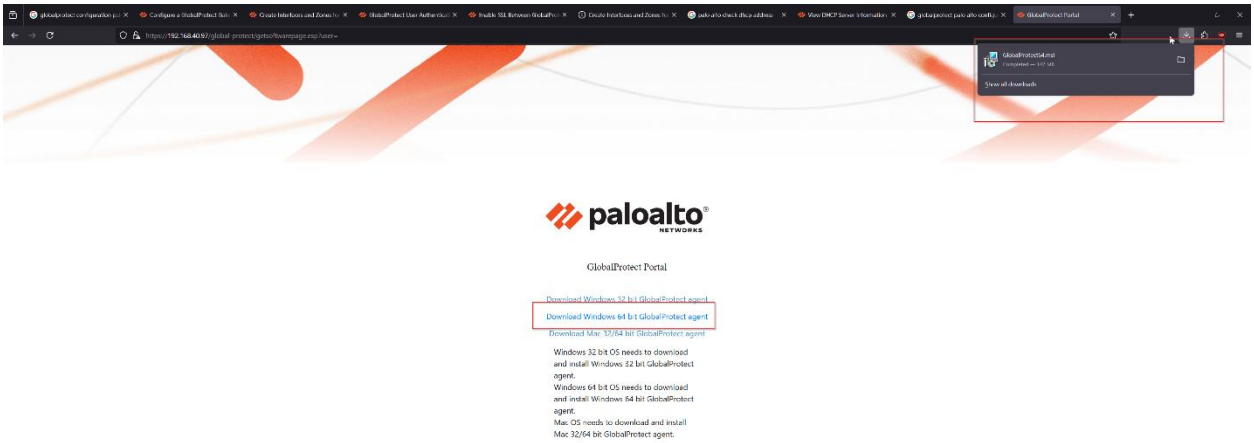


24) Input the credentials of our test user that we just created.

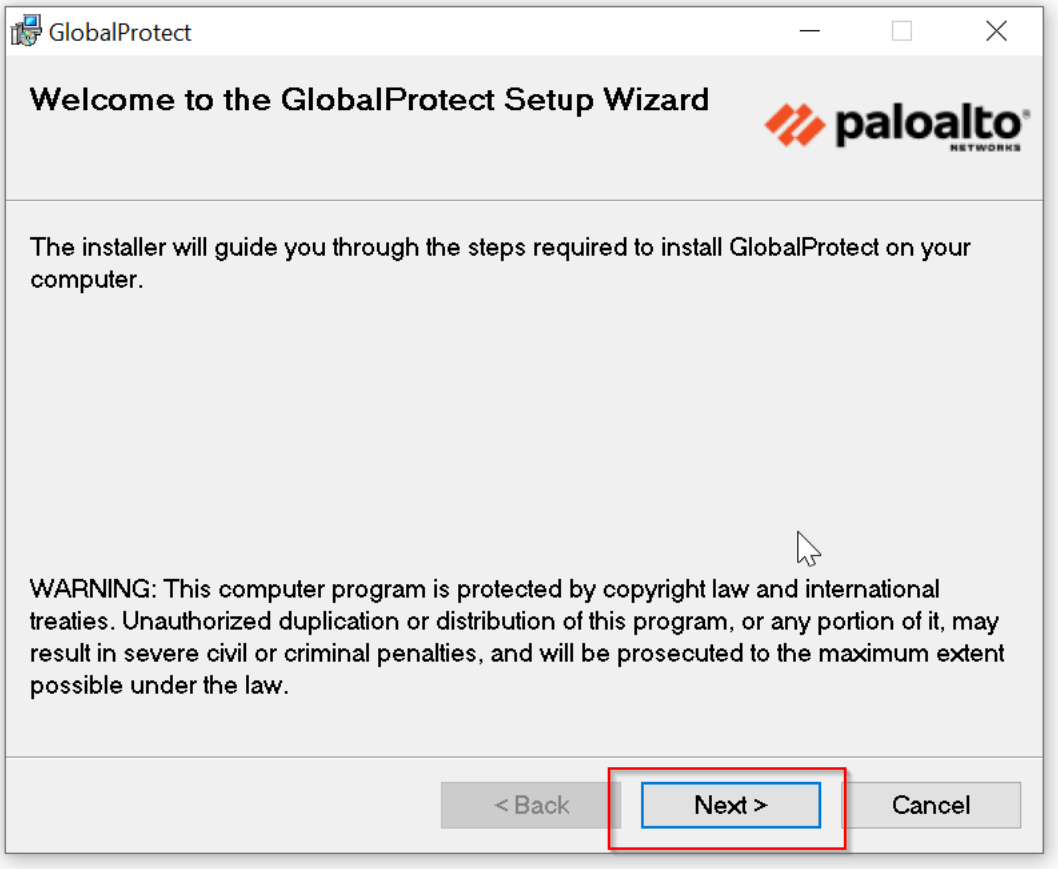
25) After a successful login, we can download our GlobalProtect VPN that is hosted by our firewall.

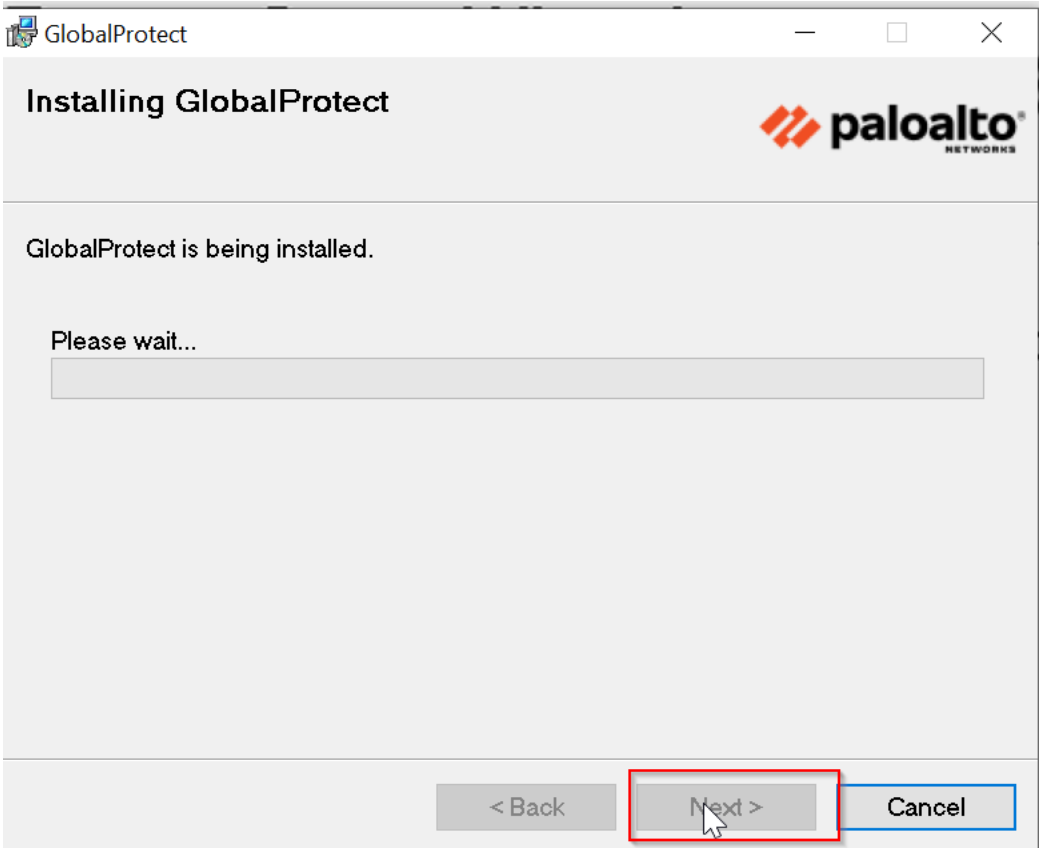


26) In this case, since we are running Windows, we should use the Windows 64 bit variant.

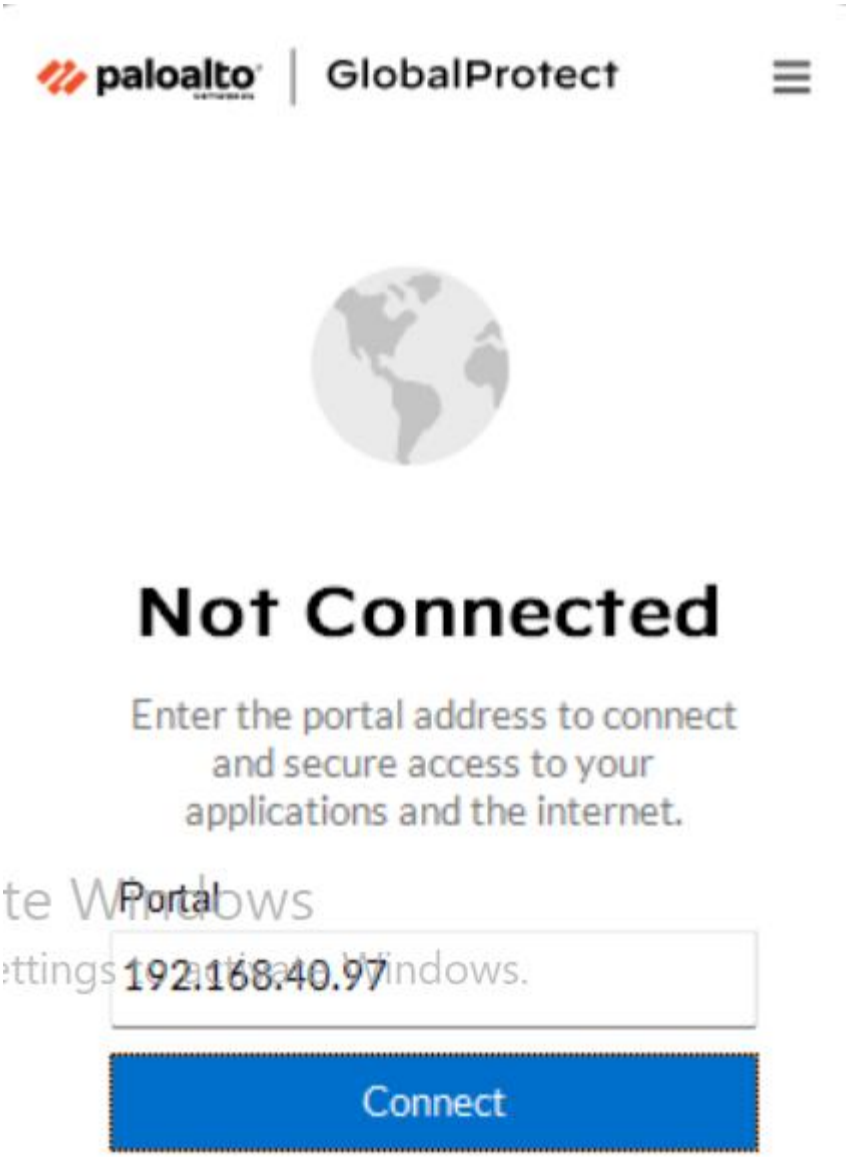


27) Download and install the GlobalProtect client.



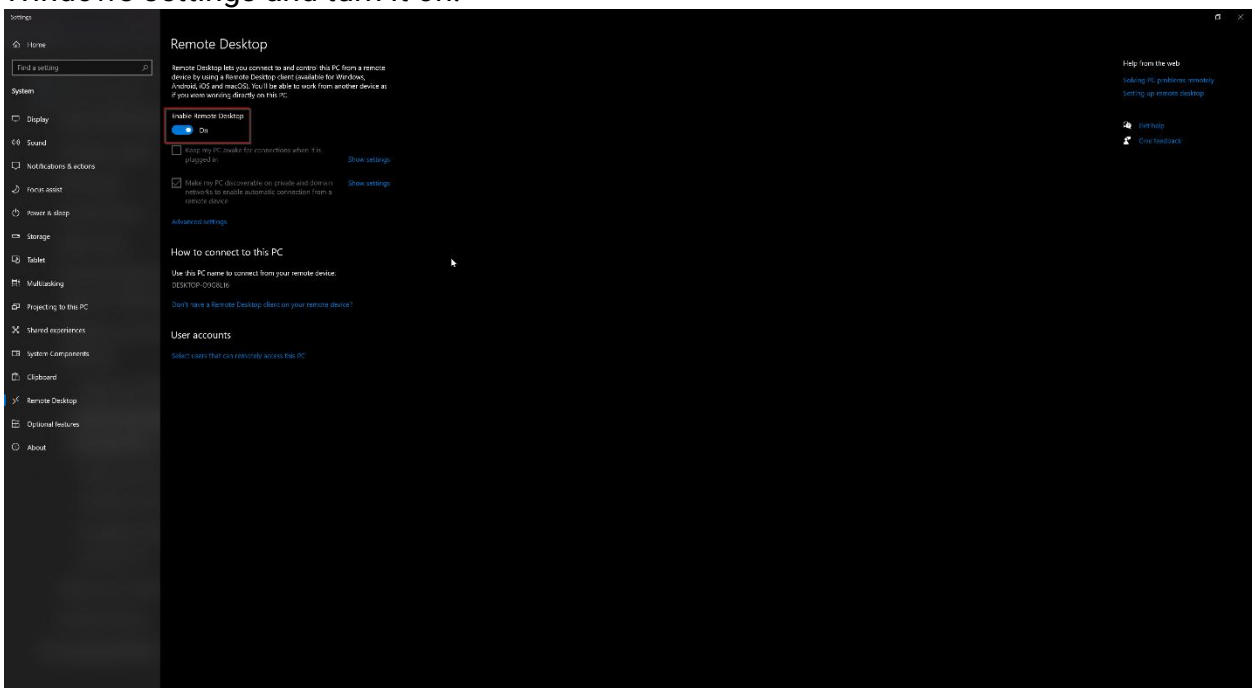

GlobalProtect
Installing GlobalProtect
paloalto
NETWORKS
GlobalProtect is being installed.
Please wait...
< Back
Next >
Cancel

28) After a successful installation, GlobalProtect client should look like this.

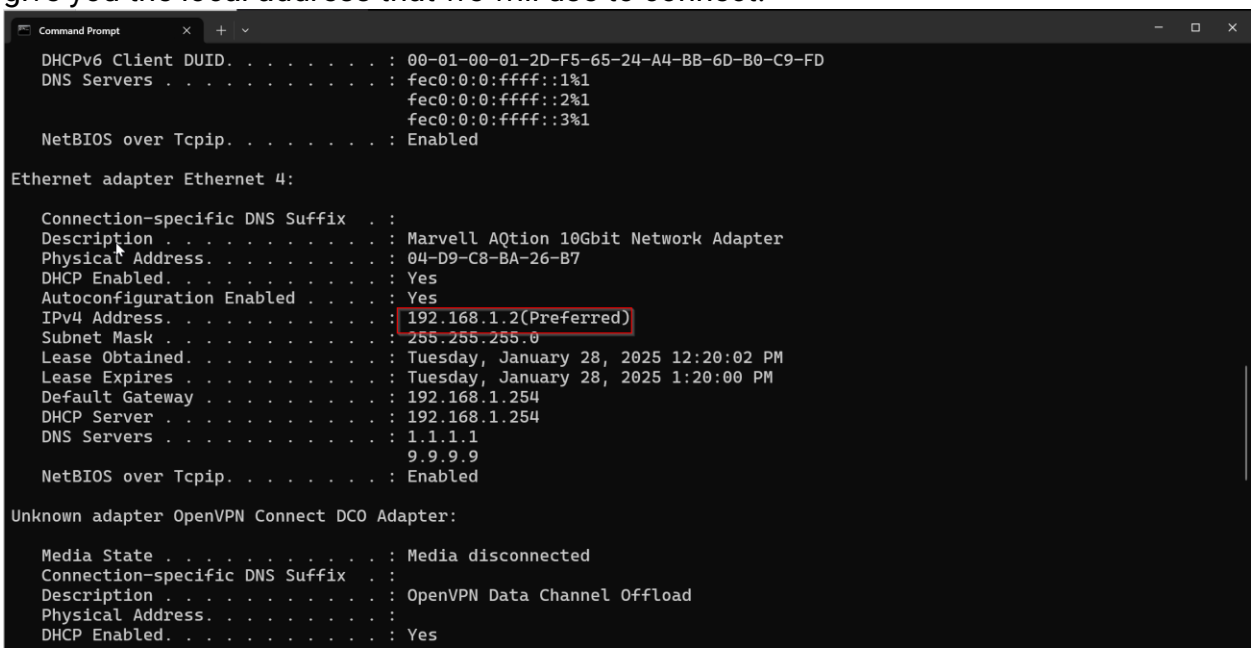


29) Our final step is to configure RDP normally, as if we were using it locally. Since the VPN will tunnel our end user into the local network, we can test our RDP with local configurations, then test it when our VPN is on. To enable RDP, navigate to the

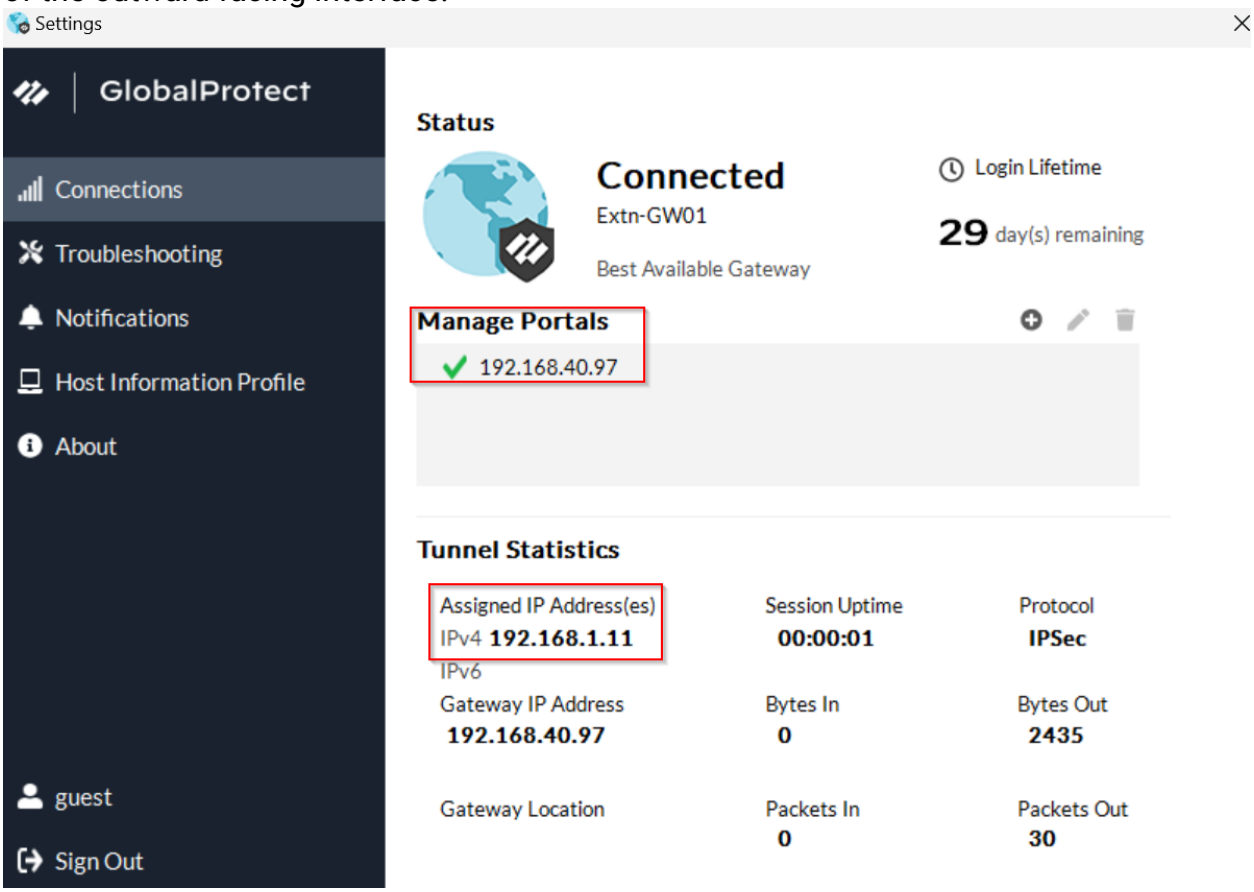
Windows settings and turn it on.



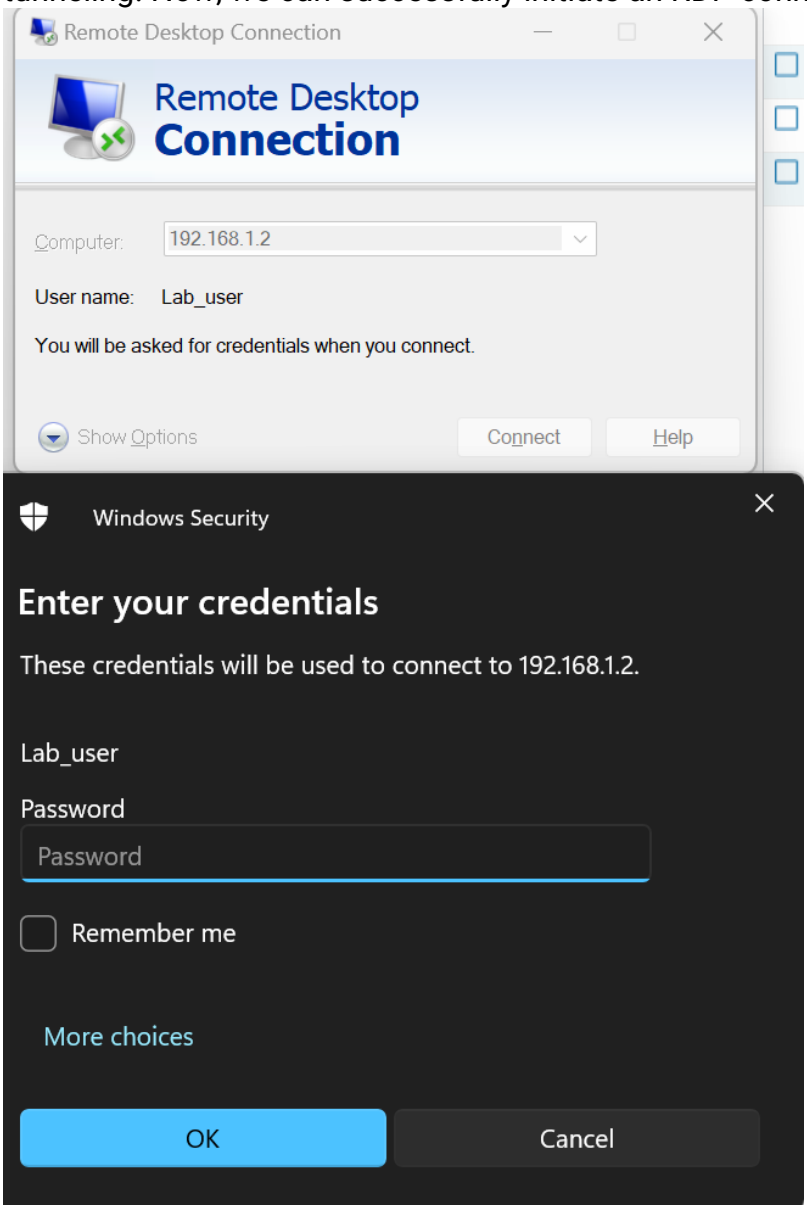
30) Find the IP address of the RDP host using “ipconfig” in a CMD terminal. This will give you the local address that we will use to connect.



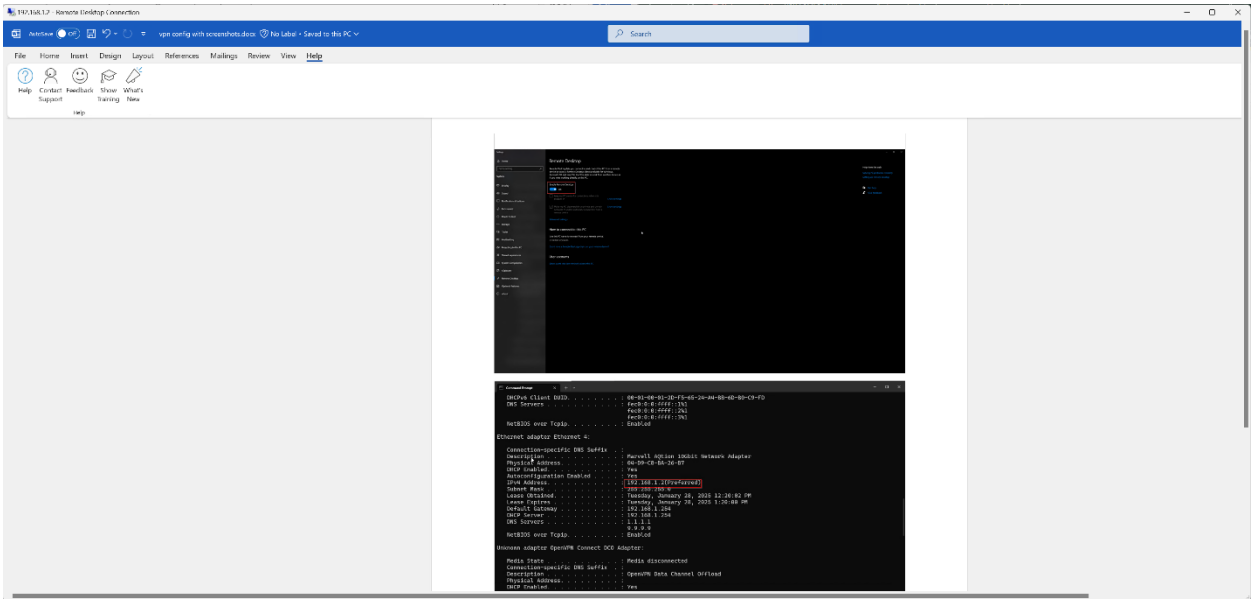
31) Now, we can enable the VPN client with the address that we previously configured of the outward facing interface.



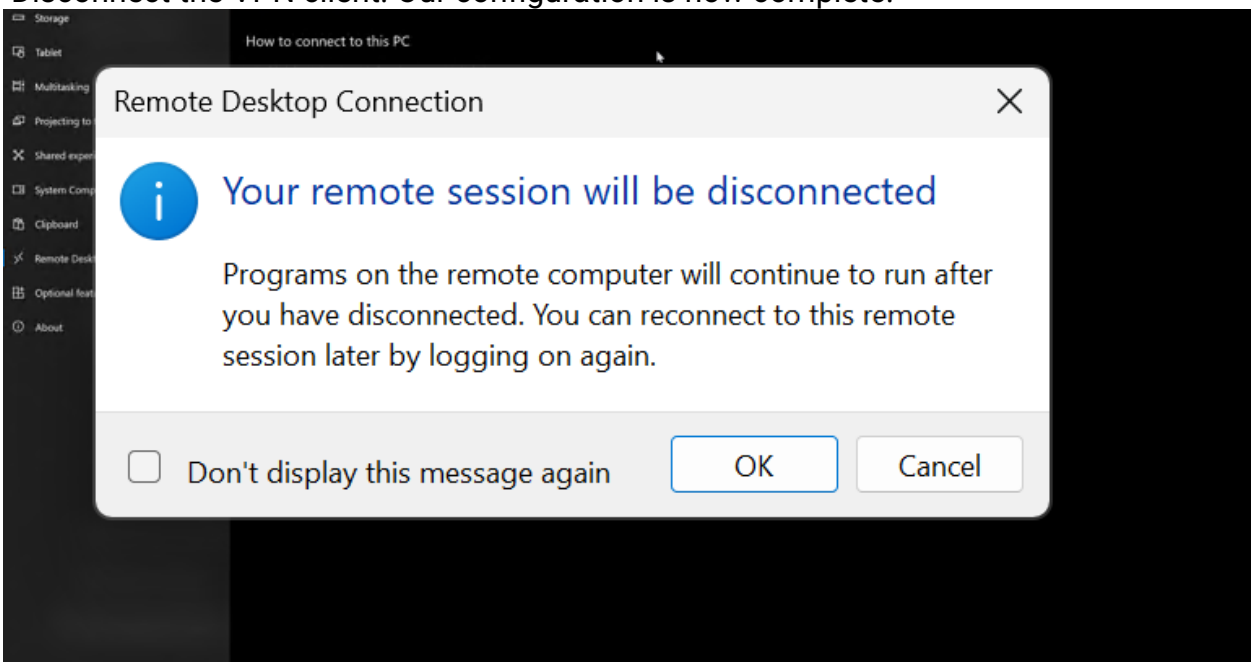
32) Notice that you receive an IP address inside the network. This is from the tunneling. Now, we can successfully initiate an RDP connection with the host.



33) Success! Note that the RDP connection works, and we are looking at the host machine.



34) Disconnect the VPN client. Our configuration is now complete.



Packet Captures:

Packet captures most accurately display the functions of our VPN. We can see that the packet capture on the interface from the external network (the side that connects into our local network via VPN) has all encrypted traffic exiting. On the other hand, captures from inside our local network on the machine hosting RDP have unencrypted traffic. Conceptually, this functions exactly as expected. Our VPN should tunnel encrypted

traffic through an external, insecure network into our internal network, where it is then unencrypted to communicate with internal services.

First, let's take a look at the external computers captures:

1	0.000000	192.168.40.128	192.168.40.97	TCP	66	62578 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.001531	192.168.40.128	192.168.40.97	TCP	54	62578 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
3	0.015007	192.168.40.128	192.168.40.97	TLSv1.2	373	Client Hello (SNI=192.168.40.97)
4	0.027508	192.168.40.128	192.168.40.97	TCP	54	62578 → 443 [ACK] Seq=320 Ack=2722 Win=131328 Len=0
5	0.029301	192.168.40.128	192.168.40.97	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
6	0.033915	192.168.40.128	192.168.40.97	TLSv1.2	85	Encrypted Alert
7	0.034168	192.168.40.128	192.168.40.97	TCP	54	62578 → 443 [FIN, ACK] Seq=477 Ack=2773 Win=131328 Len=0
8	0.035496	192.168.40.128	192.168.40.97	TCP	54	62578 → 443 [RST, ACK] Seq=478 Ack=2804 Win=0 Len=0
9	0.038806	192.168.40.128	192.168.40.97	TCP	66	62579 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
10	0.040210	192.168.40.128	192.168.40.97	TCP	54	62579 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
11	0.041703	192.168.40.128	192.168.40.97	TLSv1.2	210	Client Hello
12	0.054410	192.168.40.128	192.168.40.97	TCP	54	62579 → 443 [ACK] Seq=157 Ack=2717 Win=131328 Len=0
13	0.059682	192.168.40.128	192.168.40.97	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
14	0.067776	192.168.40.128	192.168.40.97	TLSv1.2	608	Application Data
15	0.109559	192.168.40.128	192.168.40.97	TLSv1.2	401	Application Data
16	0.109747	192.168.40.128	192.168.40.97	TLSv1.2	544	Application Data
17	0.178701	192.168.40.128	192.168.40.97	TCP	54	62579 → 443 [ACK] Seq=1674 Ack=4161 Win=129792 Len=0
18	0.492511	192.168.40.128	192.168.40.97	TCP	66	62580 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
19	0.493835	192.168.40.128	192.168.40.97	TCP	54	62580 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
20	0.509285	192.168.40.128	192.168.40.97	TLSv1.2	373	Client Hello (SNI=192.168.40.97)
21	0.525384	192.168.40.128	192.168.40.97	TCP	54	62580 → 443 [ACK] Seq=320 Ack=2722 Win=131328 Len=0
22	0.526796	192.168.40.128	192.168.40.97	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
23	0.535677	192.168.40.128	192.168.40.97	TLSv1.2	85	Encrypted Alert
24	0.535928	192.168.40.128	192.168.40.97	TCP	54	62580 → 443 [FIN, ACK] Seq=477 Ack=2773 Win=131328 Len=0
25	0.537373	192.168.40.128	192.168.40.97	TCP	54	62580 → 443 [RST, ACK] Seq=478 Ack=2804 Win=0 Len=0
26	0.540653	192.168.40.128	192.168.40.97	TLSv1.2	608	Application Data
27	0.601664	192.168.40.128	192.168.40.97	TLSv1.2	401	Application Data
28	0.601853	192.168.40.128	192.168.40.97	TLSv1.2	922	Application Data
29	0.712597	192.168.40.128	192.168.40.97	TCP	54	62579 → 443 [ACK] Seq=3443 Ack=6289 Win=130304 Len=0
30	1.118682	192.168.40.128	192.168.40.97	TCP	66	62582 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
31	1.121060	192.168.40.128	192.168.40.97	TCP	54	62582 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
32	1.121332	192.168.40.128	192.168.40.97	TLSv1.2	373	Client Hello (SNI=192.168.40.97)
33	1.134293	192.168.40.128	192.168.40.97	TCP	54	62582 → 443 [ACK] Seq=320 Ack=2722 Win=131328 Len=0
34	1.135176	192.168.40.128	192.168.40.97	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
35	1.139262	192.168.40.128	192.168.40.97	TLSv1.2	85	Encrypted Alert
36	1.139393	192.168.40.128	192.168.40.97	TCP	54	62582 → 443 [FIN, ACK] Seq=477 Ack=2773 Win=131328 Len=0
37	1.142216	192.168.40.128	192.168.40.97	TLSv1.2	601	Application Data
38	1.143419	192.168.40.128	192.168.40.97	TCP	54	62582 → 443 [RST, ACK] Seq=478 Ack=2804 Win=0 Len=0
39	1.199304	192.168.40.128	192.168.40.97	TCP	54	62579 → 443 [ACK] Seq=3990 Ack=7369 Win=131328 Len=0
40	1.312701	192.168.40.128	192.168.40.97	TLSv1.2	390	Application Data
41	1.312883	192.168.40.128	192.168.40.97	TLSv1.2	1023	Application Data
42	1.370929	192.168.40.128	192.168.40.97	TCP	54	62579 → 443 [ACK] Seq=5295 Ack=8987 Win=131328 Len=0
43	1.531986	192.168.40.128	192.168.40.97	TLSv1.2	394	Application Data
44	1.532131	192.168.40.128	192.168.40.97	TLSv1.2	527	Application Data
45	1.554348	192.168.40.128	192.168.40.97	TCP	54	62579 → 443 [ACK] Seq=6108 Ack=11174 Win=131328 Len=0

We initially see a back and forth, which is the VPN connecting. This establishes the connection through exchanges of certificates and keys, which are used for the rest of the session.

From there, all we can see is UDP traffic:

193	16.675056	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
194	16.675712	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
195	16.675874	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
196	17.601452	192.168.40.128	192.168.40.97	UDP	174	60127 → 4501 Len=132
197	17.679848	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
198	17.947024	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
199	17.947195	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
200	18.292127	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
201	18.600039	192.168.40.128	192.168.40.97	UDP	174	60127 → 4501 Len=132
202	18.888585	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
203	19.303509	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
204	19.617803	192.168.40.128	192.168.40.97	UDP	174	60127 → 4501 Len=132
205	19.618264	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
206	19.680778	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
207	19.680937	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
208	19.900260	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
209	20.308833	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
210	20.622793	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
211	20.701315	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
212	20.701478	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
213	20.905331	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
214	21.627794	192.168.40.128	192.168.40.97	UDP	174	60127 → 4501 Len=132
215	21.627965	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
216	21.628019	192.168.40.128	192.168.40.97	UDP	174	60127 → 4501 Len=132
217	21.846798	192.168.40.128	192.168.40.97	UDP	174	60127 → 4501 Len=132
218	21.973657	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
219	22.174889	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
220	22.179027	192.168.40.128	192.168.40.97	UDP	126	60127 → 4501 Len=84
221	22.199426	192.168.40.128	192.168.40.97	UDP	174	60127 → 4501 Len=132
222	22.271163	192.168.40.128	192.168.40.97	UDP	126	60127 → 4501 Len=84
223	22.318221	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
224	22.318380	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
225	22.915008	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
226	22.915188	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
227	22.978156	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
228	23.637187	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
229	23.637369	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
230	23.684406	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
231	23.684556	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
232	23.982452	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
233	25.630710	192.168.40.128	192.168.40.97	UDP	174	60127 → 4501 Len=132
234	25.630896	192.168.40.128	192.168.40.97	UDP	174	60127 → 4501 Len=132
235	25.687480	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100
236	25.693697	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
237	25.693845	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
238	25.991750	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
239	25.991918	192.168.40.128	192.168.40.97	UDP	158	60127 → 4501 Len=116
240	26.331301	192.168.40.128	192.168.40.97	UDP	142	60127 → 4501 Len=100

This continues for the rest of the session, until we finally disconnect from the VPN with a SYN ACK.

1015	41.497562	192.168.40.128	192.168.40.97	TCP	54	62601 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
1016	41.498116	192.168.40.128	192.168.40.97	TLSv1.2	373	Client Hello (SNI=192.168.40.97)
1017	41.510022	192.168.40.128	192.168.40.97	TCP	54	62601 → 443 [ACK] Seq=320 Ack=2722 Win=131328 Len=0
1018	41.511069	192.168.40.128	192.168.40.97	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
1019	41.515477	192.168.40.128	192.168.40.97	TLSv1.2	482	Application Data
1020	41.541859	192.168.40.128	192.168.40.97	TLSv1.2	85	Encrypted Alert
1021	41.542190	192.168.40.128	192.168.40.97	TCP	54	62601 → 443 [FIN, ACK] Seq=905 Ack=3458 Win=130560 Len=0
1022	41.543728	192.168.40.128	192.168.40.97	TCP	54	62601 → 443 [RST, ACK] Seq=906 Ack=3489 Win=0 Len=0
1023	53.717741	192.168.40.128	192.168.40.97	TCP	54	62579 → 443 [FIN, ACK] Seq=6108 Ack=11174 Win=131328 Len=0
1024	53.719065	192.168.40.128	192.168.40.97	TCP	54	62579 → 443 [ACK] Seq=6109 Ack=11175 Win=131328 Len=0

This UDP traffic does not show labelled as RDP, even though we know that it is. This is because it is encrypted. RDP itself runs off UDP in order for fastest speed, which is how we know that this traffic is the RDP connection. This proves that the VPN is correctly

encrypting our traffic. A middle man would not be able to decrypt this traffic without the necessary certificates and keys, ensuring security in the connection.

Now, let's take a look at how this differs from the traffic inside the network. We see the unencrypted version now:

1	0.000000	192.168.1.11	192.168.1.2	TCP	66	64634 → 3389 [SYN] Seq=0 Win=65280 Len=0 MSS=1360 WS=256 SACK_PERM
2	0.000236	192.168.1.2	192.168.1.11	TCP	66	3389 → 64634 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 WS=1 SACK_PERM
3	0.004457	192.168.1.11	192.168.1.2	TCP	60	64634 → 3389 [ACK] Seq=1 Ack=1 Win=262400 Len=0
4	0.006800	192.168.1.11	192.168.1.2	RDP	100	Cookie: mstshash=Lab_user, Negotiate Request
5	0.021825	192.168.1.2	192.168.1.11	RDP	73	Negotiate Response
6	0.075723	192.168.1.11	192.168.1.2	TCP	60	64634 → 3389 [ACK] Seq=47 Ack=20 Win=262400 Len=0
7	3.482199	192.168.1.11	192.168.1.2	TLSv1.2	331	Client Hello (SNI=192.168.1.2)
8	3.482911	192.168.1.2	192.168.1.11	TLSv1.2	900	Server Hello, Certificate, Server Hello Done
9	3.488903	192.168.1.11	192.168.1.2	TLSv1.2	372	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	3.492567	192.168.1.2	192.168.1.11	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
11	3.536052	192.168.1.11	192.168.1.2	TLSv1.2	140	Application Data
12	3.537081	192.168.1.2	192.168.1.11	TLSv1.2	345	Application Data
13	3.545467	192.168.1.11	192.168.1.2	TLSv1.2	726	Application Data
14	3.548729	192.168.1.2	192.168.1.11	TLSv1.2	142	Application Data
15	3.552954	192.168.1.11	192.168.1.2	TCP	60	64634 → 3389 [RST, ACK] Seq=1400 Ack=1296 Win=0 Len=0
16	3.601951	192.168.1.11	192.168.1.2	TCP	66	64635 → 3389 [SYN] Seq=0 Win=65280 Len=0 MSS=1360 WS=256 SACK_PERM
17	3.602345	192.168.1.2	192.168.1.11	TCP	66	3389 → 64635 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 WS=1 SACK_PERM
18	3.604935	192.168.1.11	192.168.1.2	TCP	60	64635 → 3389 [ACK] Seq=1 Ack=1 Win=262400 Len=0
19	3.609979	192.168.1.11	192.168.1.2	RDP	100	Cookie: mstshash=Lab_user, Negotiate Request
20	3.631096	192.168.1.2	192.168.1.11	RDP	73	Negotiate Response
21	3.648276	192.168.1.11	192.168.1.2	TLSv1.2	331	Client Hello (SNI=192.168.1.2)
22	3.649060	192.168.1.2	192.168.1.11	TLSv1.2	900	Server Hello, Certificate, Server Hello Done
23	3.653016	192.168.1.11	192.168.1.2	TLSv1.2	372	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
24	3.656445	192.168.1.2	192.168.1.11	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
25	3.660314	192.168.1.11	192.168.1.2	TLSv1.2	140	Application Data
26	3.661205	192.168.1.2	192.168.1.11	TLSv1.2	345	Application Data
27	3.670168	192.168.1.11	192.168.1.2	TLSv1.2	726	Application Data
28	3.673340	192.168.1.2	192.168.1.11	TLSv1.2	142	Application Data
29	3.677103	192.168.1.11	192.168.1.2	TLSv1.2	169	Application Data
30	3.678104	192.168.1.2	192.168.1.11	TLSv1.2	87	Application Data
31	3.682165	192.168.1.11	192.168.1.2	TLSv1.2	545	Application Data
32	3.684801	192.168.1.2	192.168.1.11	TLSv1.2	209	Application Data
33	3.687197	192.168.1.11	192.168.1.2	TLSv1.2	95	Application Data

Note that the RDP packets are now labelled, and that application data is sent unencrypted. This is expected, since we are capturing data from the local side, after the VPN packets have already been decrypted by our firewall.

Finally, we see a stream of RDP packets, mirroring those of the UDP packets we previously saw. The only difference is that this time, they are unencrypted.

55	3.742725	192.168.1.11	192.168.1.2	RDPUDP	1274	SYN,CORRELATIONID,SYNEX
56	3.743521	192.168.1.2	192.168.1.11	RDPUDP	1274	SYN,SYNEX
57	3.767322	192.168.1.11	192.168.1.2	TLSv1.2	302	Client Hello
58	3.767322	192.168.1.11	192.168.1.2	RDPUDP2	1049	AOA,DUMMY
59	3.775125	192.168.1.2	192.168.1.11	RDPUDP2	1060	ACK,OVERHEAD,DELAYACK,AOA,DUMMY
60	3.775217	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
61	3.775262	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
62	3.775309	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
63	3.775373	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
64	3.775423	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
65	3.775476	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
66	3.775520	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
67	3.775564	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
68	3.775605	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
69	3.775648	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
70	3.775692	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
71	3.775736	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
72	3.775783	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
73	3.775830	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
74	3.775873	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
75	3.775919	192.168.1.2	192.168.1.11	RDPUDP2	1052	DELAYACK,AOA,DUMMY
76	3.775965	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
77	3.776011	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
78	3.776055	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
79	3.776096	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
80	3.776142	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
81	3.776294	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
82	3.776346	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
83	3.776398	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
84	3.776547	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
85	3.776594	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
86	3.776639	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
87	3.776806	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
88	3.776850	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
89	3.776899	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
90	3.776945	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
91	3.776993	192.168.1.2	192.168.1.11	RDPUDP2	1052	DELAYACK,AOA,DUMMY
92	3.777040	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY
93	3.777082	192.168.1.2	192.168.1.11	RDPUDP2	1049	AOA,DUMMY

As we can see, the VPN functions as expected, encrypting traffic between an unsecured network in order to connect to another, local network in a secure manner.

Problems:

1) **Not Downloading PA GlobalProtect Agent Files**
   When we first configuring the VPN, we mistakenly thought that the actual GlobalProtect client files would come pre-installed and available to install on our portal page. However, this is not the case. The ".msi" files need to be retrieved from the Palo Alto servers, from which they can then be downloaded on end user devices through the portal. This mistake has us debugging for a while, because the links on our portal would always download a dummy file. Initially, we thought

this was because of an incorrect configuration with the GlobalProtect portal, thus leaving us guessing what the issue.

2) **“Fake” Public Address – Internal Lab Network**
   Since this lab is done a “shared” setting – meaning that we are emulating a production environment inside our class – we have to use an internal address as our “public” address. Due to the number of groups in our class, as well as security concerns, it would be impossible to assign everybody a public address for testing. Because of this, our teacher has a LAN that all groups should consider as a “public” network, simulating the broader internet. Due to not understanding this difference at first, we didn’t know what address to put as our outward facing portal.

Conclusion:

This lab is a great introduction to understanding both how VPN’s work and why they work. In order to configure the VPN server, we first needed to understand the principles behind a VPN, and how they tie together. Furthermore, this lab did a good job at introducing the GlobalProtect VPN service, and how to deploy it in an industry setting.

Fortinet SOHO Firewall Configuration on FortiGate 40F

Ryan Chen

Cybersecurity

Purpose:

This lab is primarily used as a starter for familiarization to the Fortinet interface as well as FortiGate configurations. Given that SOHO is a relatively easy task to setup, this is a good time to understand how Fortinet configurations work at a basic level.

Aside from Fortinet, SOHO itself is a simple lab that provides wireless connectivity to clients at a small scale, hence the name “Small Office Home Office”. This style of configuration is common for personal networks and small enterprise networks. The primary purpose of this lab is to complete basic configuration of this type of network by setting up two SSIDs: one running WPA2 Personal, and the other running WPA2 Enterprise.

Background Information:

Barebones configuration is commonly referred to as a SOHO or Small Office/Home Office setup. Traditionally, this means that the network will be used at a small scale for a limited number of clients. As a result, the config is robust at a small scale, and easy to setup. Despite these advantages, SOHO networks are often limited in future growth and must be upgraded to support anything larger scale.

Before utilizing a firewall, configuration must be done in order to prepare the device for use by clients. This involves setting up Dynamic Host Configuration Protocol (DHCP) to automatically provide client machines with IP addresses, which ensures that all devices can reach to the internet.

Our lab makes use of an upstream internet provider that already has DHCP leasing system set up for its clients. Thus, as a client, we can configure the firewall to receive a DHCP address on a specified interface, and then internally host a DHCP server to provide local clients with internet access. This would mimic a normal scenario where an ISP gives out a DHCP address for your upstream connection.

Leasing addresses is the act of giving out a specific IP address from a pool of addresses for clients connected. These addresses have a “lifespan” attribute attached to them, and after the time is up, they will need to renew another address. This ensures addresses are always moving in circulation, which is optimal in a network where devices are constantly connecting and disconnecting to the internet, such as a SOHO network.

To follow best practices, we should also configure our zones accordingly, as we would have done on Palo Alto. Like configuring Security Zones on PA220 firewalls, Fortinet offers a Firewall Policy section in its configuration that allows us to set rules for SSIDs, interfaces, and much more. For the purposes of this lab, we all traffic to flow bidirectionally.

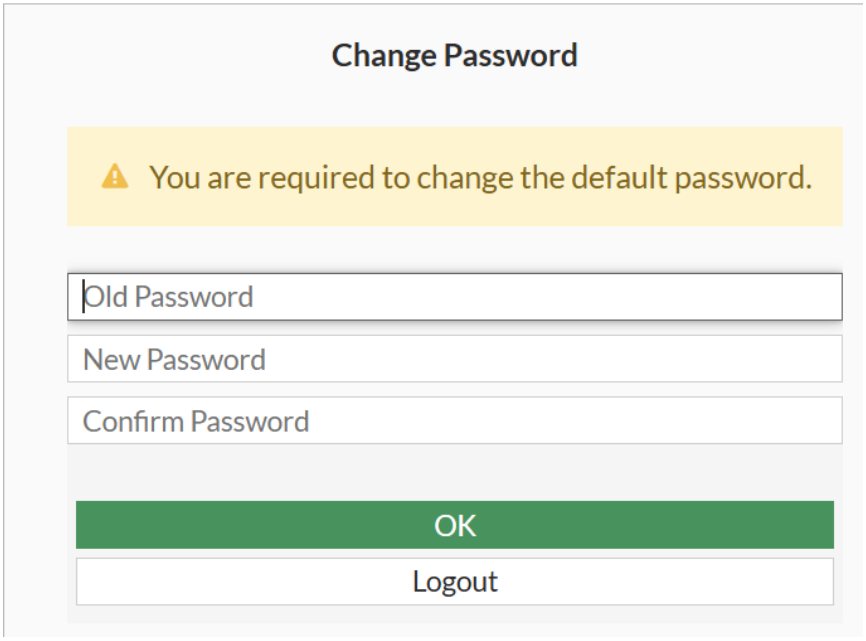
One of the primary differences between Fortinet access points and other traditional access points like Cisco ones is the ease of configuration. On Fortinet devices, access points can be configured directly by accessing the GUI of the main firewall. This provides much more accessibility because when a WAP is connected into the network of a Fortinet firewall, the firewall will be automatically detected and available for configuration. This means that instead of having to manage each AP individually over webpages, the APs can all be managed through a central firewall and GUI.

Finally, we use a local authentication setup instead of RADIUS for the WPA2 Enterprise setup used. This reduces a lot of the configuration overhead necessary for getting WPA2 Enterprise up and running, making the process a lot more simple.

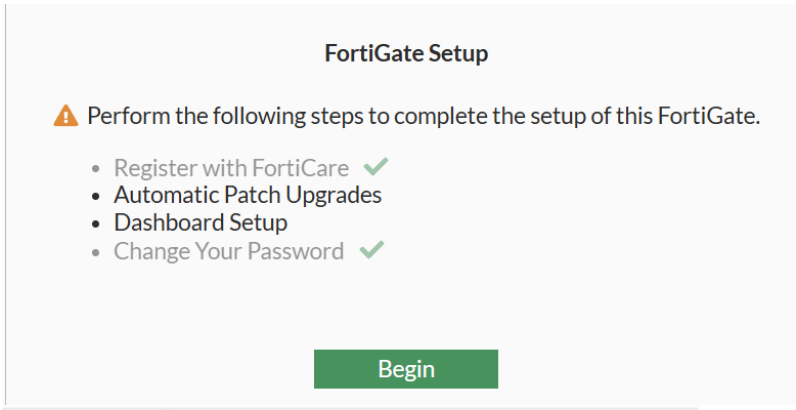
Lab Summary:

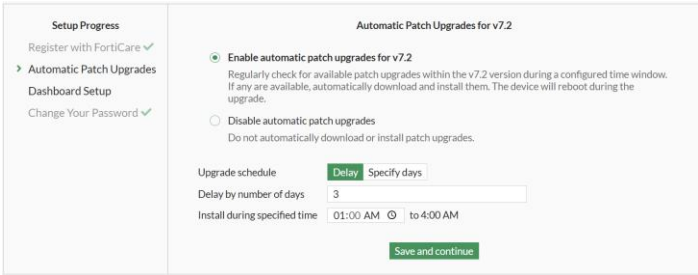
1) Head to https://192.168.1.99 to access the firewall GUI.

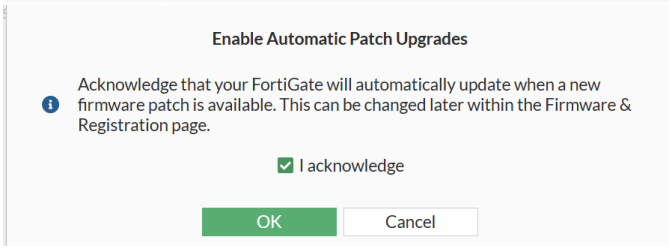
2) Enter “admin” for the username and press enter. The firewall should be wiped meaning that everything is fresh. Now, we can set our password.



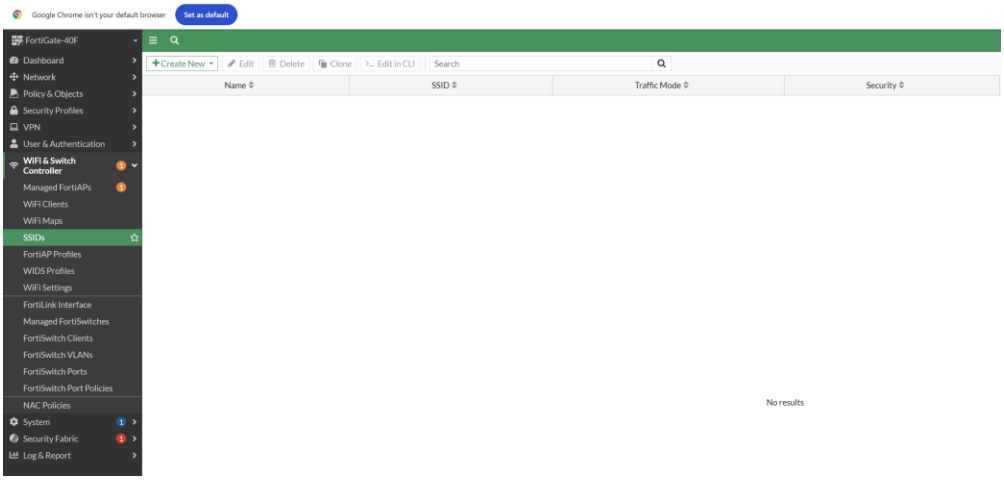
3) After setting a password, begin basic setup. This will take care of initialization of the device and other housekeeping items.



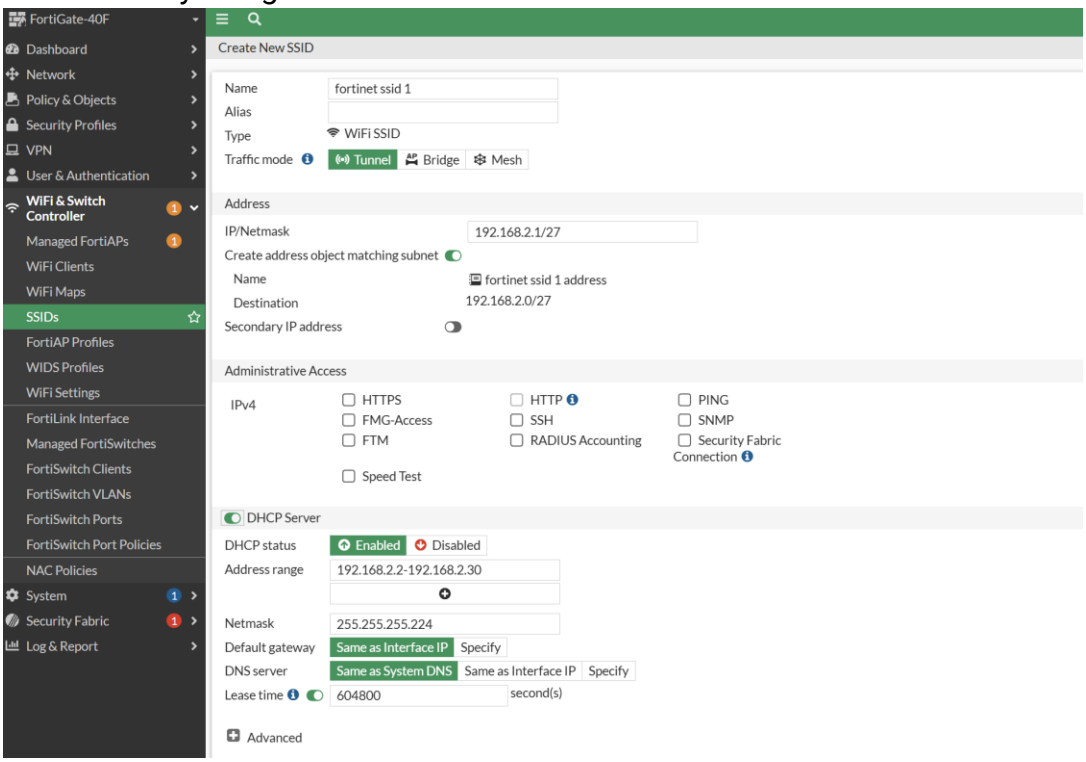




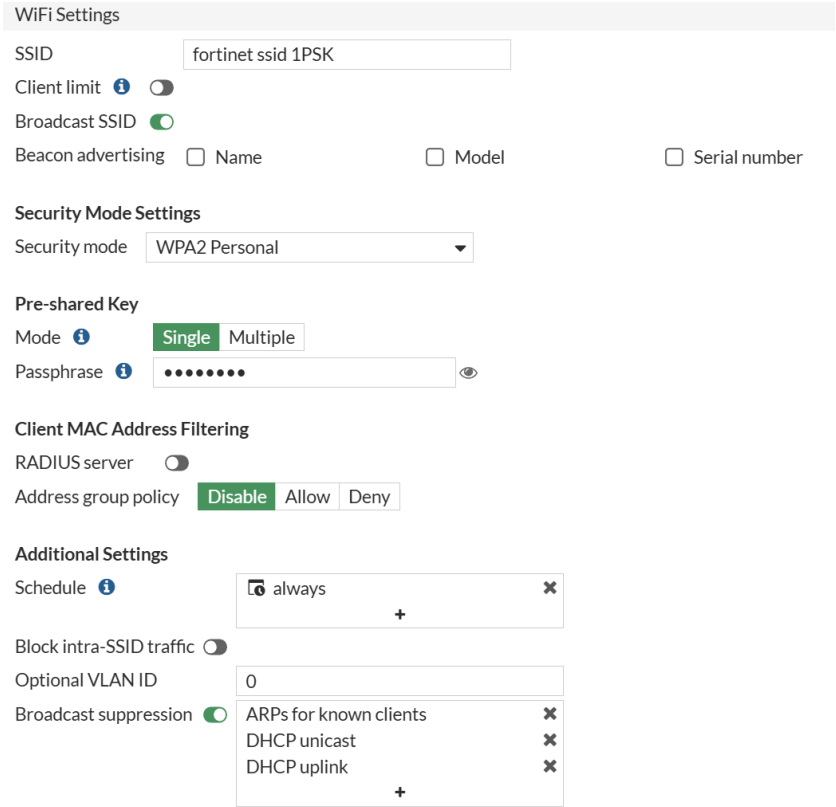
4) Once logged into the GUI, go to “Wifi & Switch Controller > SSIDs”



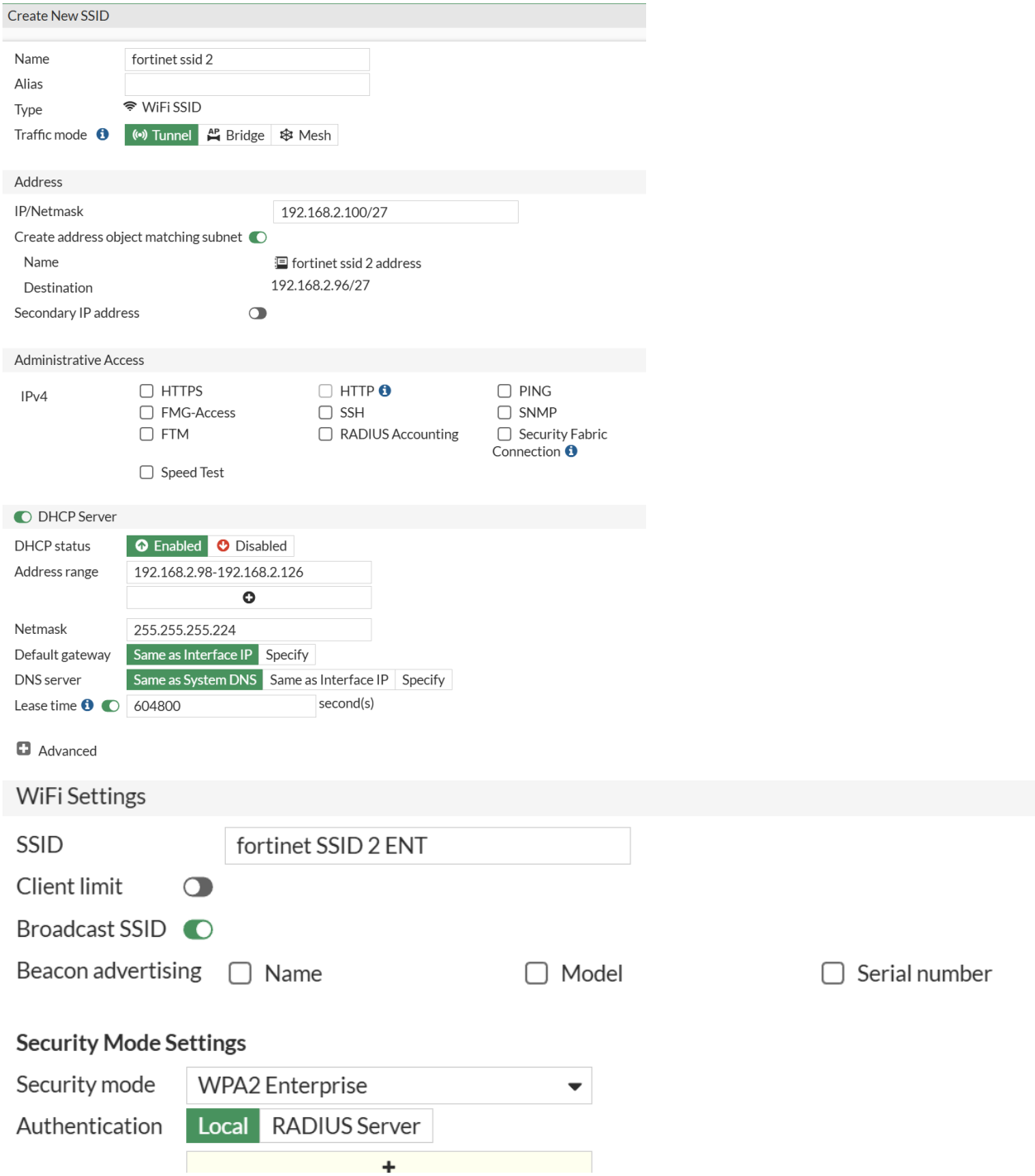
5) Create the first SSID. Add a name, and an IP/netmask that is different from the LAN IP address. This is to prevent overlapping ranges that will render the device unreachable. Also, make sure to turn on the DHCP server in order to automatically assign host IP addresses.



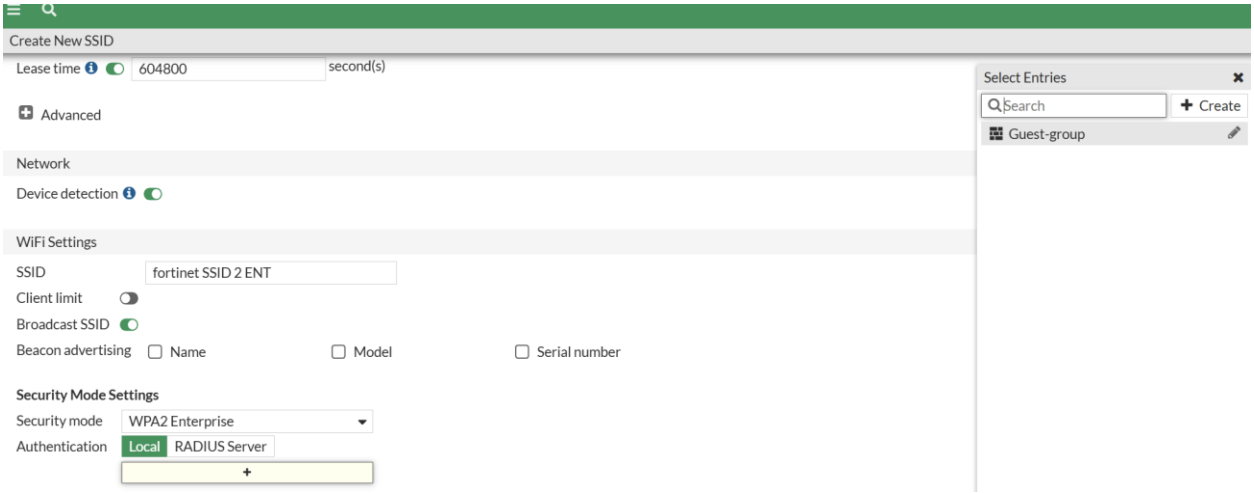
6) Set the SSID name, as well as the passphrase. Our first SSID will be WPA2 Personal, so select that item.

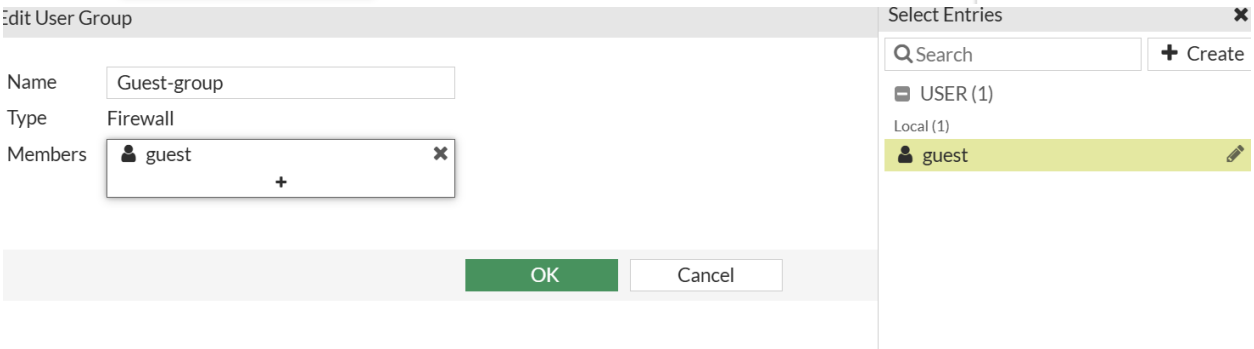


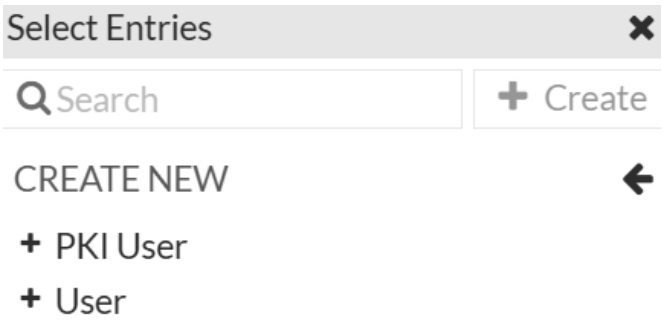
7) Now, we want to create the WPA2 Enterprise. Create another SSID and fill out the same boxes. This time, select Enterprise as the security mode.



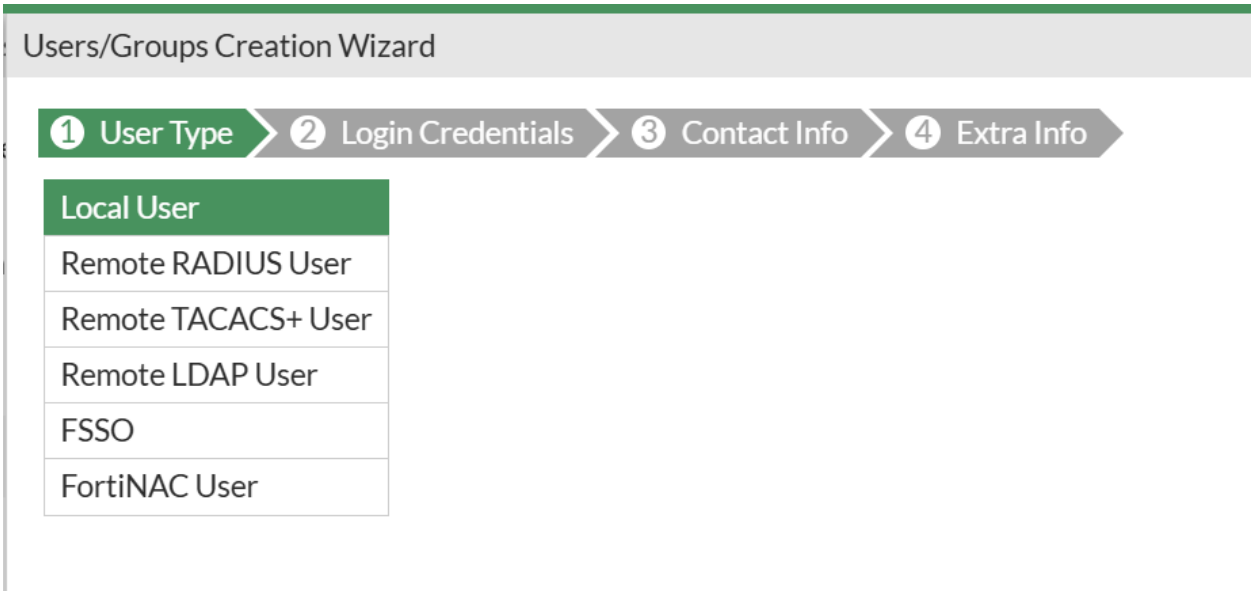
8) Now, we have to create a user for authentication purposes. Click the plus icon in order to create a new user, and create a new user group.



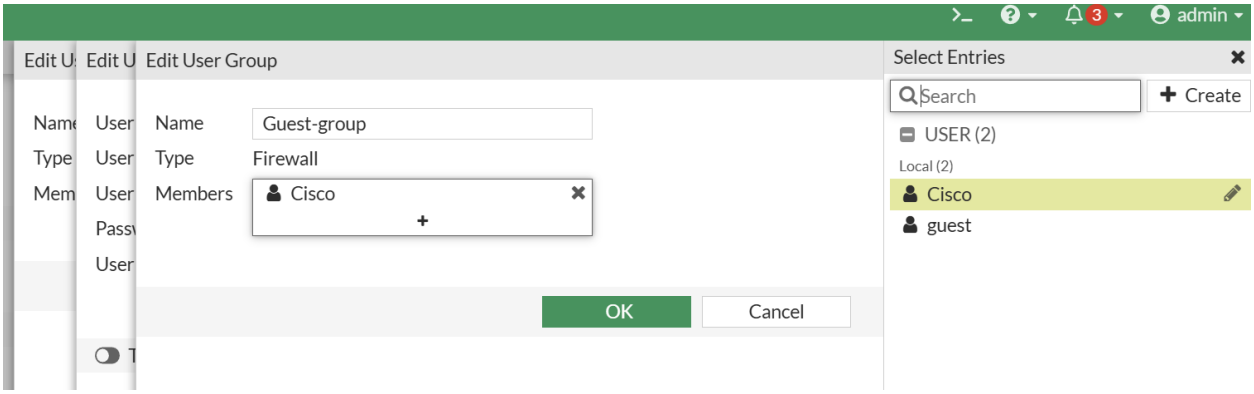




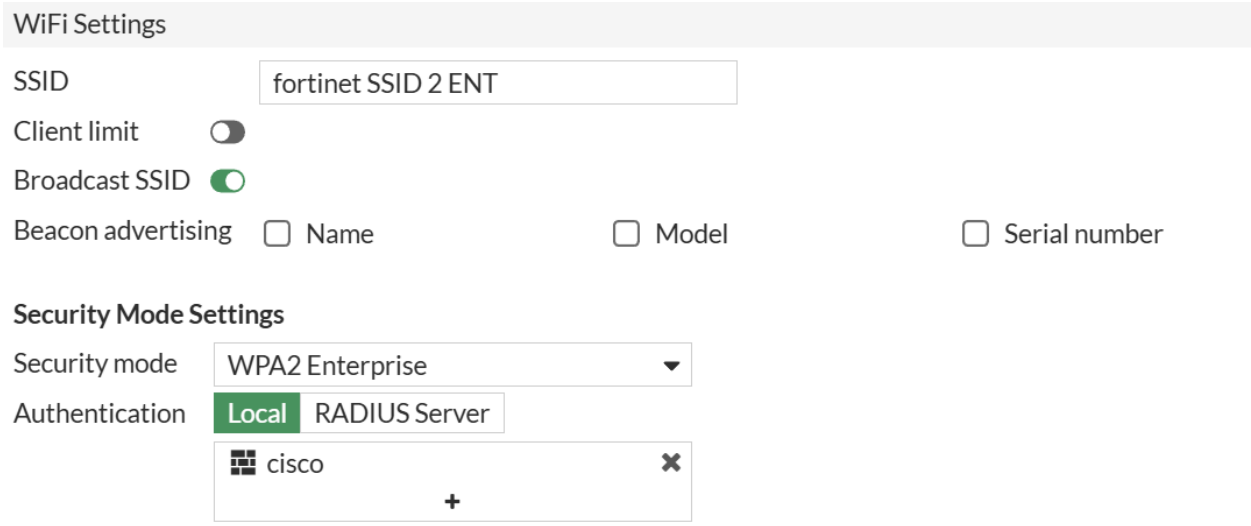
9) We want to create a locally authenticated user, so select “Local”. Now, input a username and password.


Users/Groups Creation Wizard
1 User Type
2 Login Credentials
3 Contact Info
4 Extra Info
Local User
Remote RADIUS User
Remote TACACS+ User
Remote LDAP User
FSSO
FortiNAC User

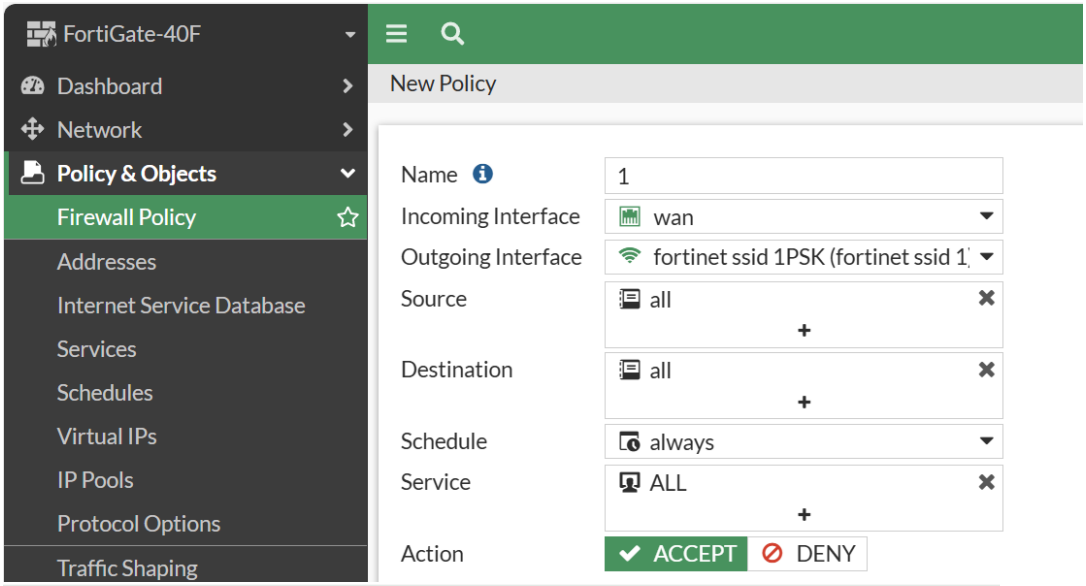
Users/Groups Creation Wizard
User Type
2 Login Credentials
3 Contact Info
4 Extra Info
Username
Cisco
Password
< Back
Next
Cancel

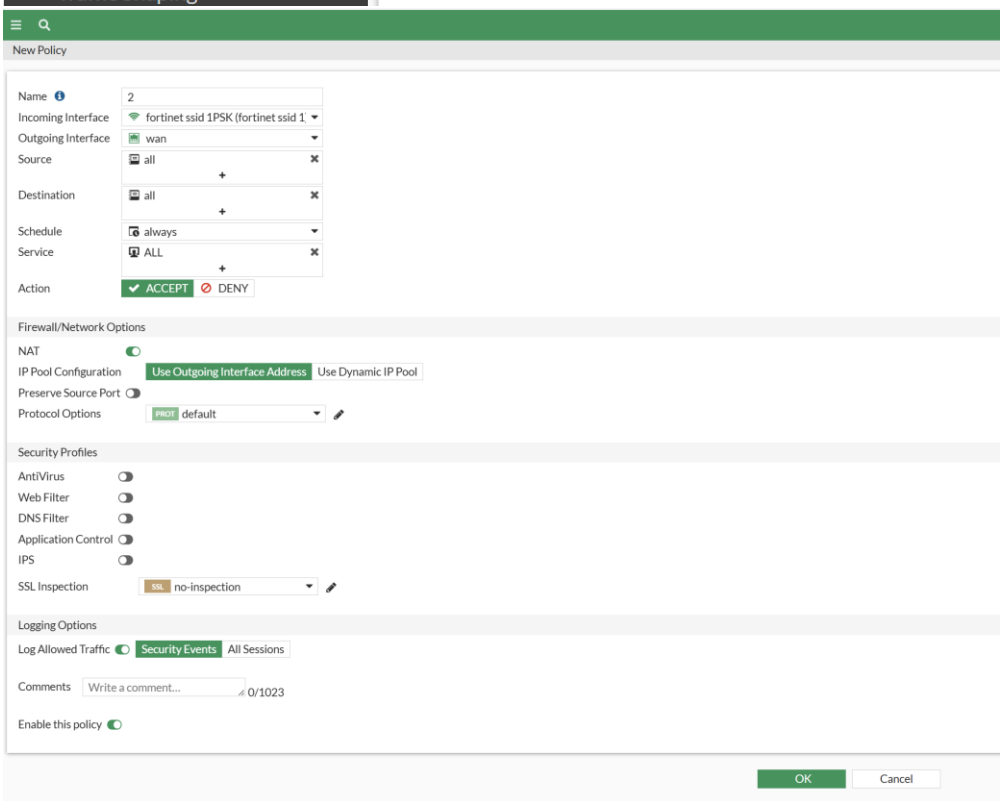


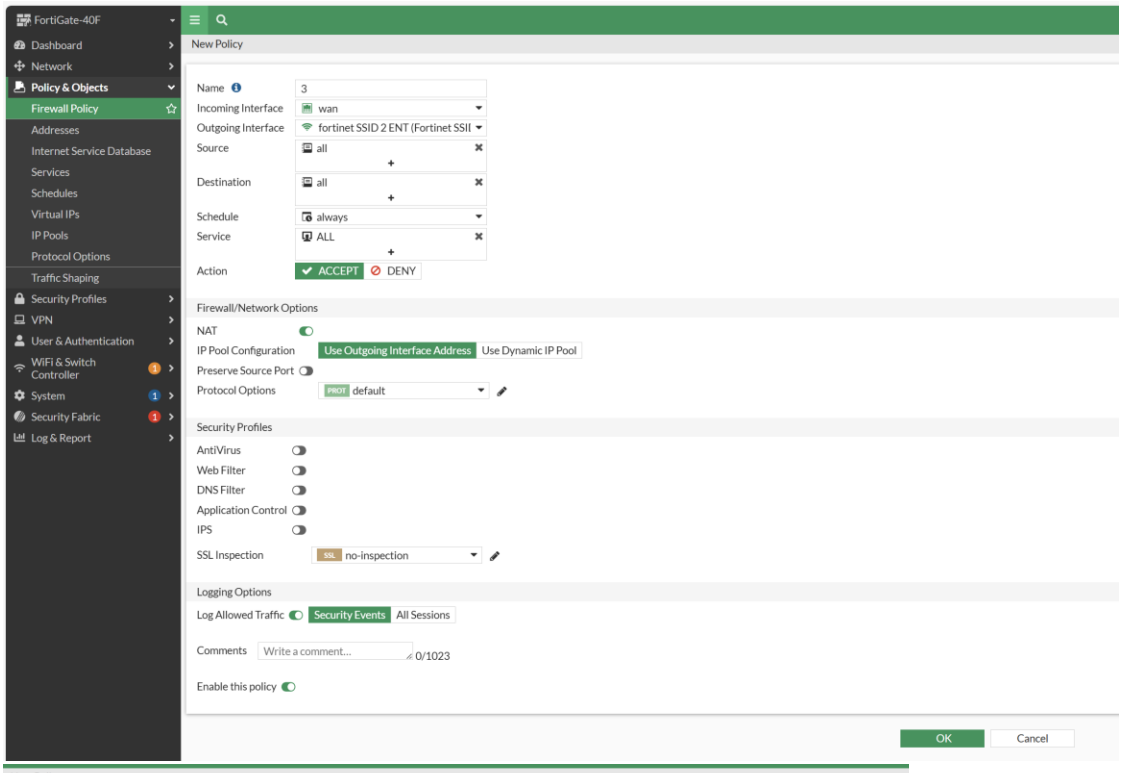
10) Upon successfully creating the User Group and local authentication user, we can assign it to the Enterprise network. This will ensure that only that user's credentials can be used to sign in.

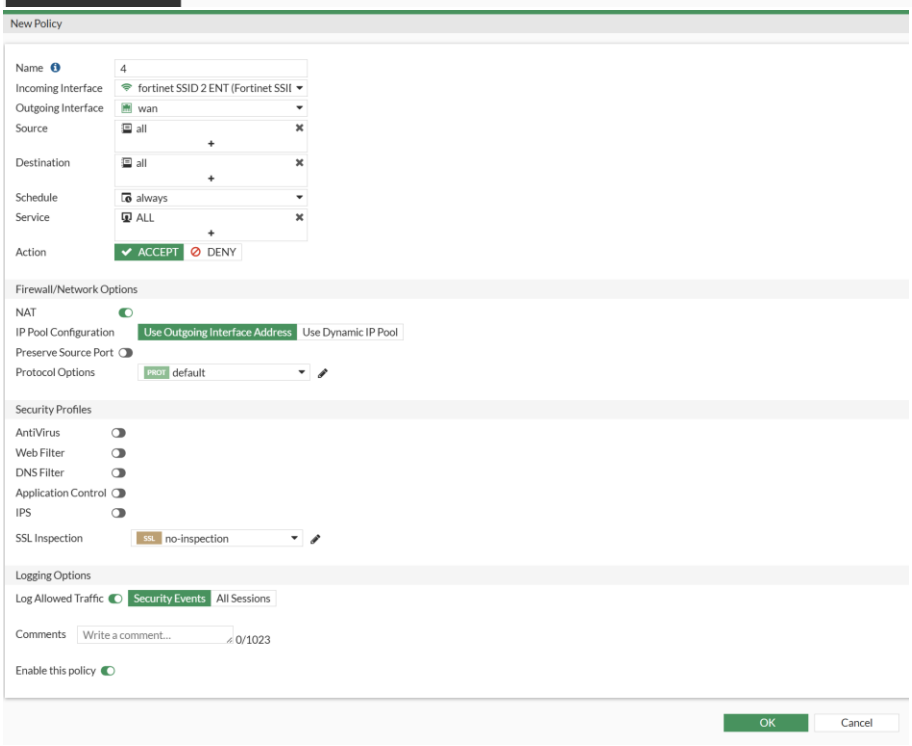


11) Now, all we have to do is add a firewall policy to allow traffic to flow through our network. This is important, as traffic is typically blocked for security reasons. Navigate to "Policy & Object > Firewall Policy" and create a new policy.

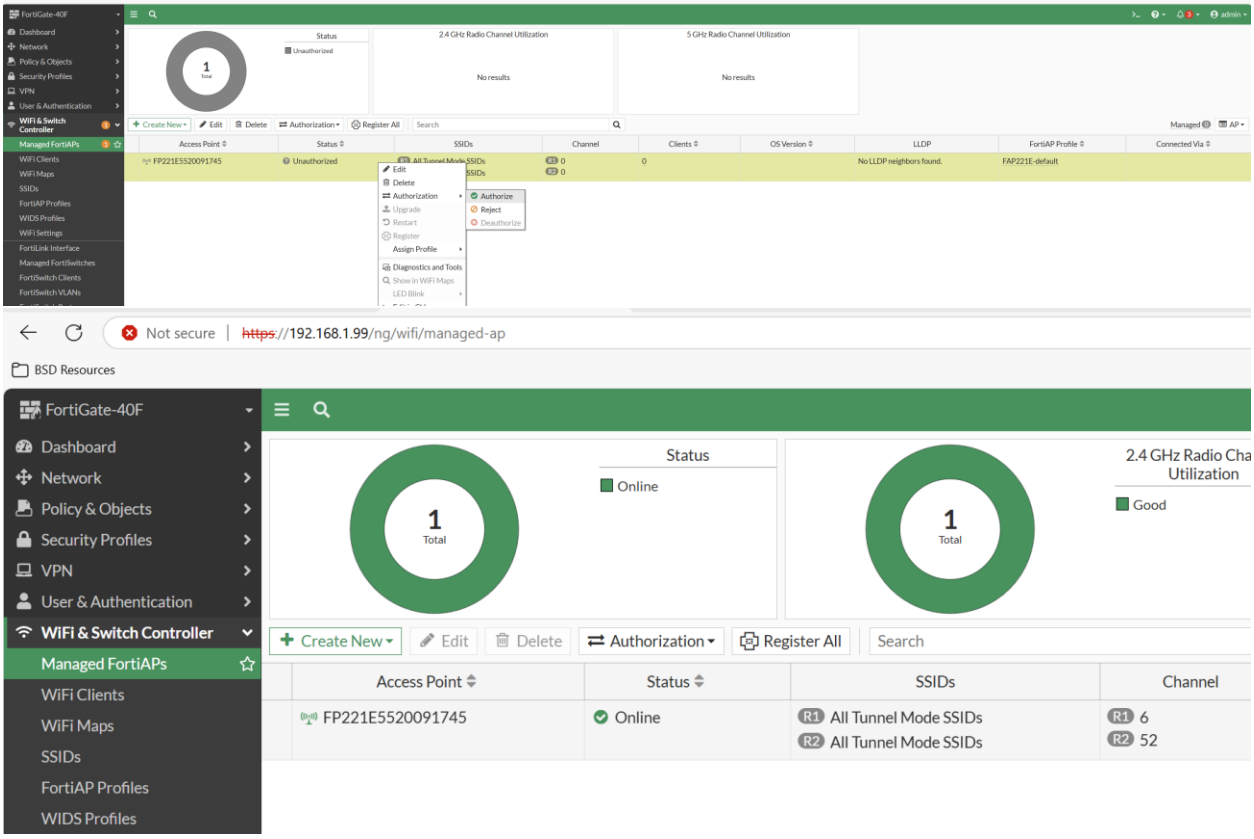

FortiGate-40F
Dashboard
Network
Policy & Objects
Firewall Policy
Addresses
Internet Service Database
Services
Schedules
Virtual IPs
IP Pools
Protocol Options
Traffic Shaping
New Policy
Name
1
Incoming Interface
wan
Outgoing Interface
fortinet ssid 1PSK (fortinet ssid 1
Source
all
Destination
all
Schedule
always
Service
ALL
Action
ACCEPT
DENY


New Policy
Name
2
Incoming Interface
fortinet ssid 1PSK (fortinet ssid 1
Outgoing Interface
wan
Source
all
Destination
all
Schedule
always
Service
ALL
Action
ACCEPT
DENY
Firewall/Network Options
NAT
IP Pool Configuration
Use Outgoing Interface Address
Use Dynamic IP Pool
Preserve Source Port
Protocol Options
default
Security Profiles
AntiVirus
Web Filter
DNS Filter
Application Control
IPS
SSL Inspection
no-inspection
Logging Options
Log Allowed Traffic
Security Events
All Sessions
Comments
Write a comment...
0/1023
Enable this policy
OK
Cancel

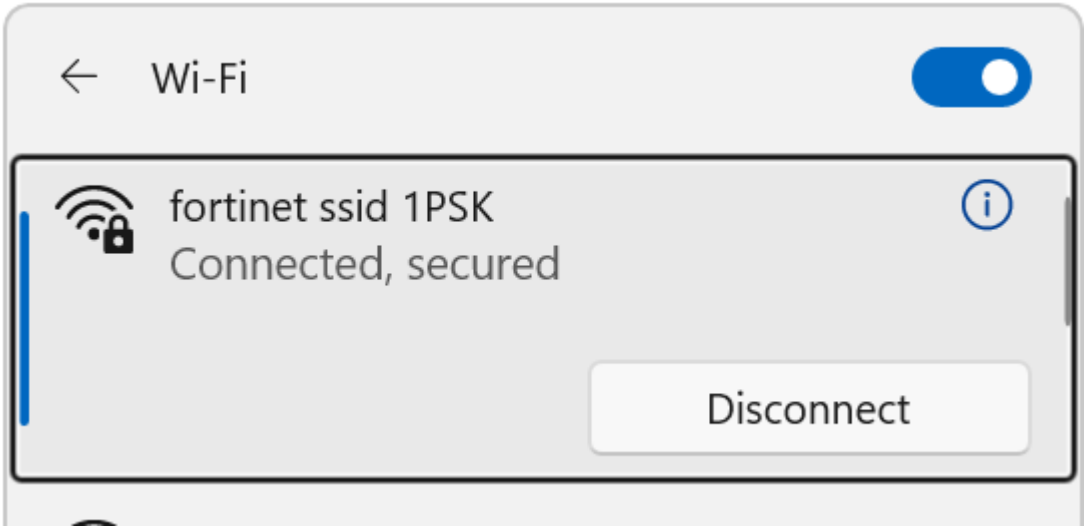




12) After creating the necessary firewall policies, our last step is to authroize the Fortigate AP. Navigate to “Wifi & Switch controller > Managed Fortinet APs”, and right click on the connected AP to authorize. If you haven’t connected an AP at this stage, connect it to one of the firewall’s ethernet ports and wait for it to show up.

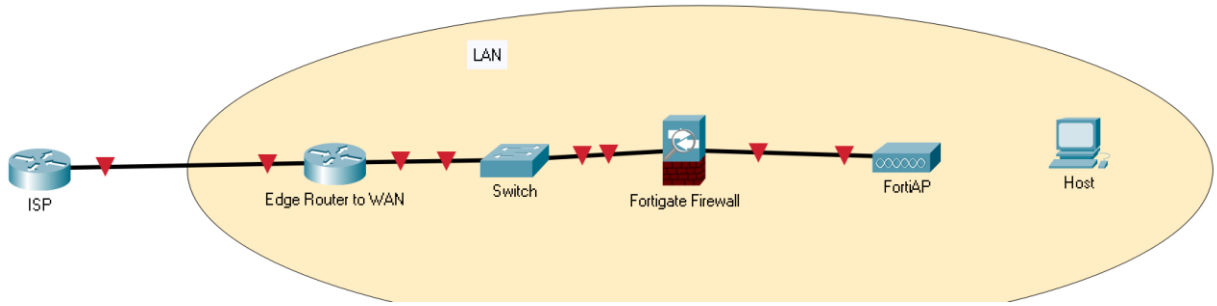


13) After our firewall shows up as “Online”, our network should be up and running. Get on a device near the access point and try and connected.



14) Success! We have configured our WPA2 Personal and WPA2 Enterprise networks.

Network Diagram:

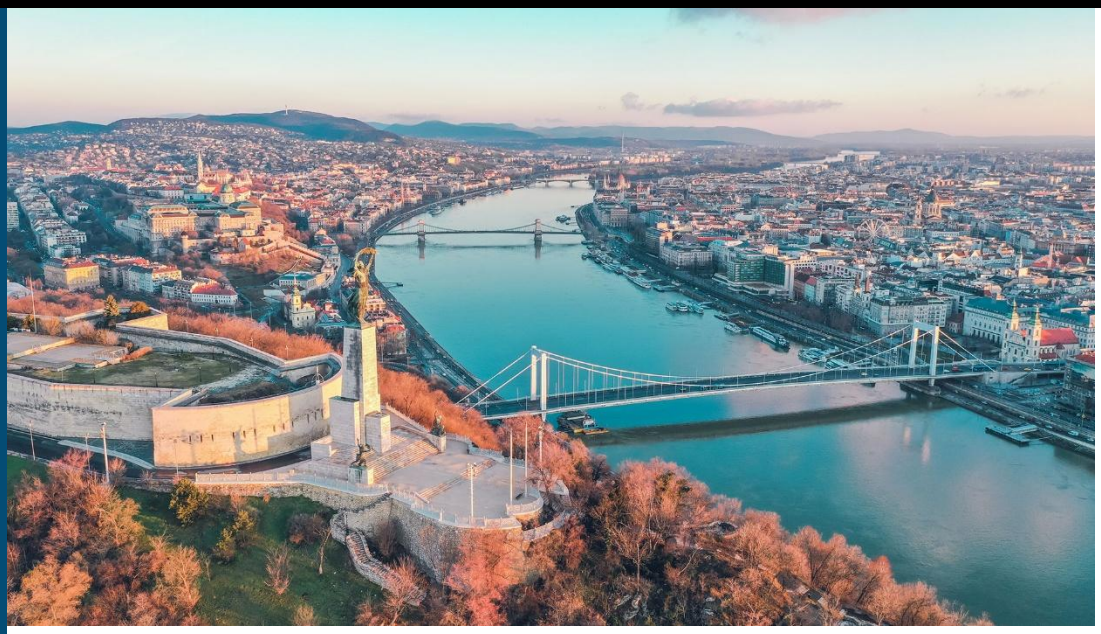


Problems:

3) **SSID Networks Requiring Firewall Policy**
   At first, we were not aware that both networks require firewall policies in order to allow network traffic to flow through. Because of this, we mistakenly thought that our configuration was entirely wrong, causing us to debug for quite a while. Furthermore, we had an issue where our WPA2 Personal network was running smoothly with a correctly configured firewall policy, but our WPA2 Enterprise network could not access any internet. This was fixed by changing our firewall policy to the appropriate values.
4) **FortiGate AP Load Times/Auto Configuration**
   Though we knew that the AP would automatically negotiate a connection with the access point, we thought that it would be quite fast. For some reason, our access point took a long time to connect, and as a result left us confused. We initially believed it was a configuration issue, only to later see that the access point was up and running with no issues.
5) **FortiGate AP POE Connector**
   Because of our unique rack setup with the ground racks, we were assigned a rack that only had normal Layer 2 ethernet switches. Unfortunately, the FortiGate AP's only ran over barrel jack connections or POE. Due to a limited supply of barrel jack connections, we were forced to try and use POE. However, with the lack of POE on our rack's switch, we needed to use a POE injector in order to supply power, and then route the ethernet connection back to our Layer 2 switch and into our firewall. This was the only way to get power to the AP, as well as connect it to the firewall.

Conclusion:

Overall, this lab was a great introduction to the utilities and GUI of Fortinet and its FortiGate firewall/AP combination network setup. We were able to successfully configure an introductory SOHO network that can be practically used in all small-scale enterprise or home network situations. This was also a good refresher on basic network configuration and best security practices for firewalls.



SSL-VPN on FortiGate 40F

Ryan Chen

Cybersecurity

Purpose:

The primary purpose of this lab is to configure and test an SSL VPN on a generic Fortinet Fortigate 40F firewall. Building off of previously learned concepts in configuring Palo Alto devices, this lab translates past information and applies it towards Fortinet devices. Furthermore, this configuration continues to utilize basic SOHO and Fortinet configuration knowledge learned in previous labs.

Background Information:

A VPN, or virtual private network, is a commonly used service that provides a multitude of different privacy, security, and convenience functions. At its core, they work by encrypting traffic between an end-user and the server, allowing a secure connection to be established from anywhere. Given that this traffic is encrypted, it is essentially improbable for an adversary to read it. Even if they do manage to get hold of the traffic by sniffing it off the network, they will be unable to read it without the key.

For this reason, VPN's are commonly used to secure traffic from an unsecure network into a secure network. By establishing this secure connection, attacks like a MITM (Man in the Middle) become less of a threat, since traffic is encrypted. Likewise, people also use VPN's for their ability to "tunnel" traffic. This refers to the ability to establish a connection that emulates the device directly on a local network. This is useful for many different use cases, but commonly required by companies when employees are doing remote work. In order to keep company traffic secure even from an unsecure network, companies can configure a VPN to tunnel employee traffic back into the network. This allows employees to work remotely on company networks, all while keeping traffic secure. In our test case, this is the primary usage that we aim to target with our configuration.

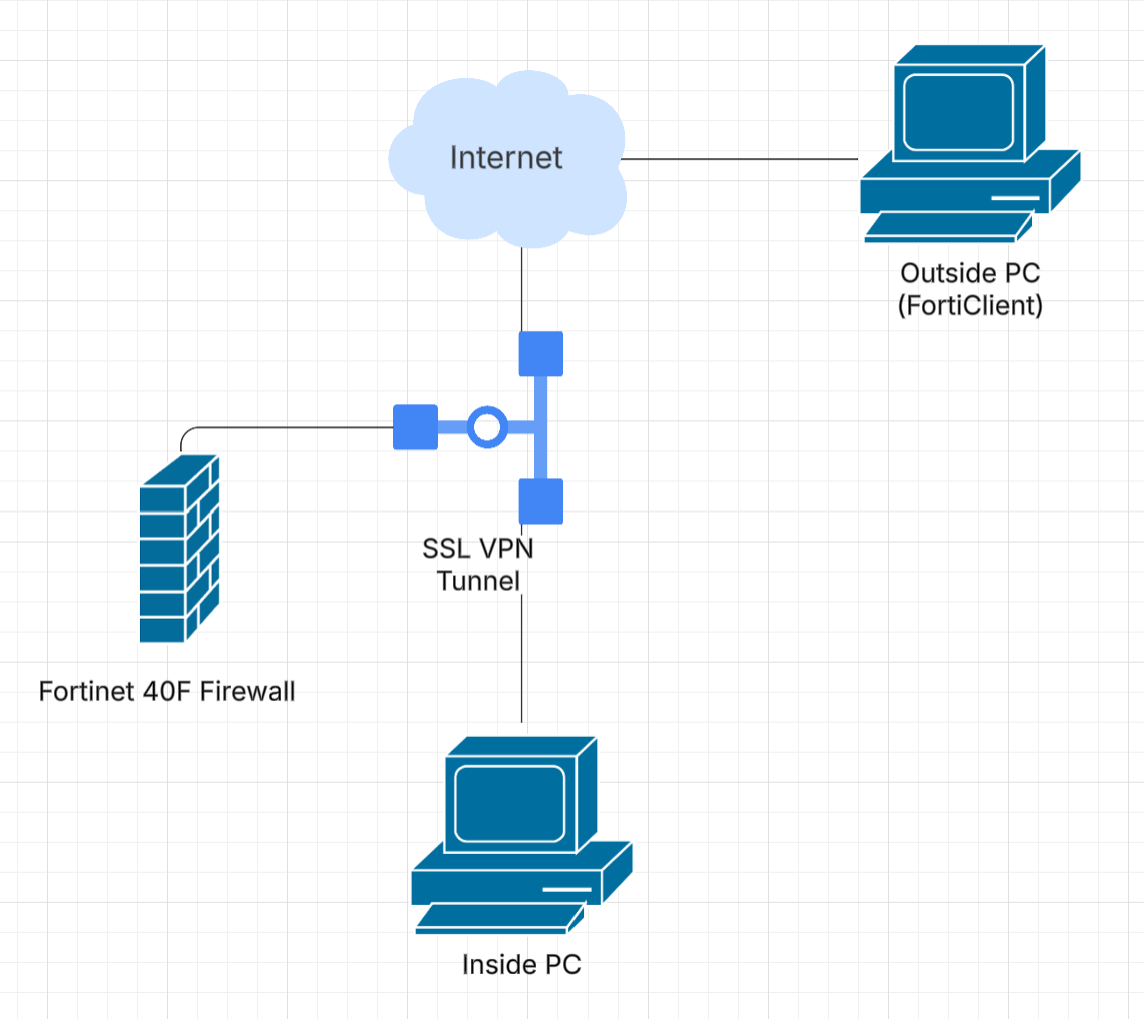
Now that we know the basic use cases of a VPN, we can start to cover how it works at a base level. Typically for a VPN to successfully send and receive traffic, end users must connect through an application configured by the VPN provider. This application works by establishing a secure connection through the use of certificates and encryption, allowing end user traffic to be encrypted in transit and decrypted by the VPN at the end.

However, this lab is specifically targeted towards configuring an SSL VPN, which functions using the Secure Socket Layer (SSL) protocol. This protocol, which is most commonly seen in securing browser web traffic, can also be used to secure VPN data. Because of the ubiquitous nature of browsers, SSL can be found on almost any device, and thus is a suitable way to implement the VPN. It simplifies VPN connections to allow for much easier configuration and usage.

SSL VPN's can provide two main features: browser-based access, and tunnel-based access. Both use the SSL protocol to send data, but function in different ways. Browser based access generates a session directly in the browser, giving secure access to a webpage through the session. This simplifies deployment, but also limits applications that can run within the web interface. On the other hand, tunnel mode creates an encrypted tunnel as if the client machine were on the local network of the SSL VPN server. This typically allows full access and much finer configuration controls. This lab focuses on tunnel based access through Fortinet's proprietary VPN client, FortiClient VPN.

Finally, one of the key concepts of this lab is remote access. For this purpose, we use the RDP (Remote Desktop Protocol) provided by Windows. This protocol allows us to configure one machine inside the network as a server, and the other machine as a end user that will ultimately establish a connection to the server to demonstrate that the VPN is working. RDP gives us the ability to remotely view another Windows desktop, a common use case for companies that want to enable employees to do remote work. We demonstrate functionality of the VPN through the FortiClient connection, and then through RDP.

Network Diagram:



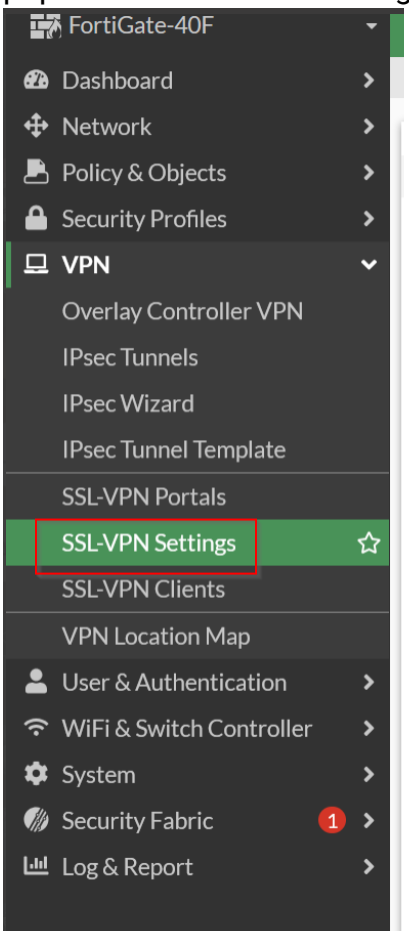
Lab Summary:

SSL VPN configuration is relatively painless on the Fortinet firewall, especially with the help of the easily navigable GUI. First, we have to define an SSL VPN configuration where we can set connection port, name, etc. From there, we simply create a firewall policy to allow the SSL VPN configuration to pass through our firewall, and our configuration is complete. Fortinet makes the SSL VPN as painless to setup as possible.

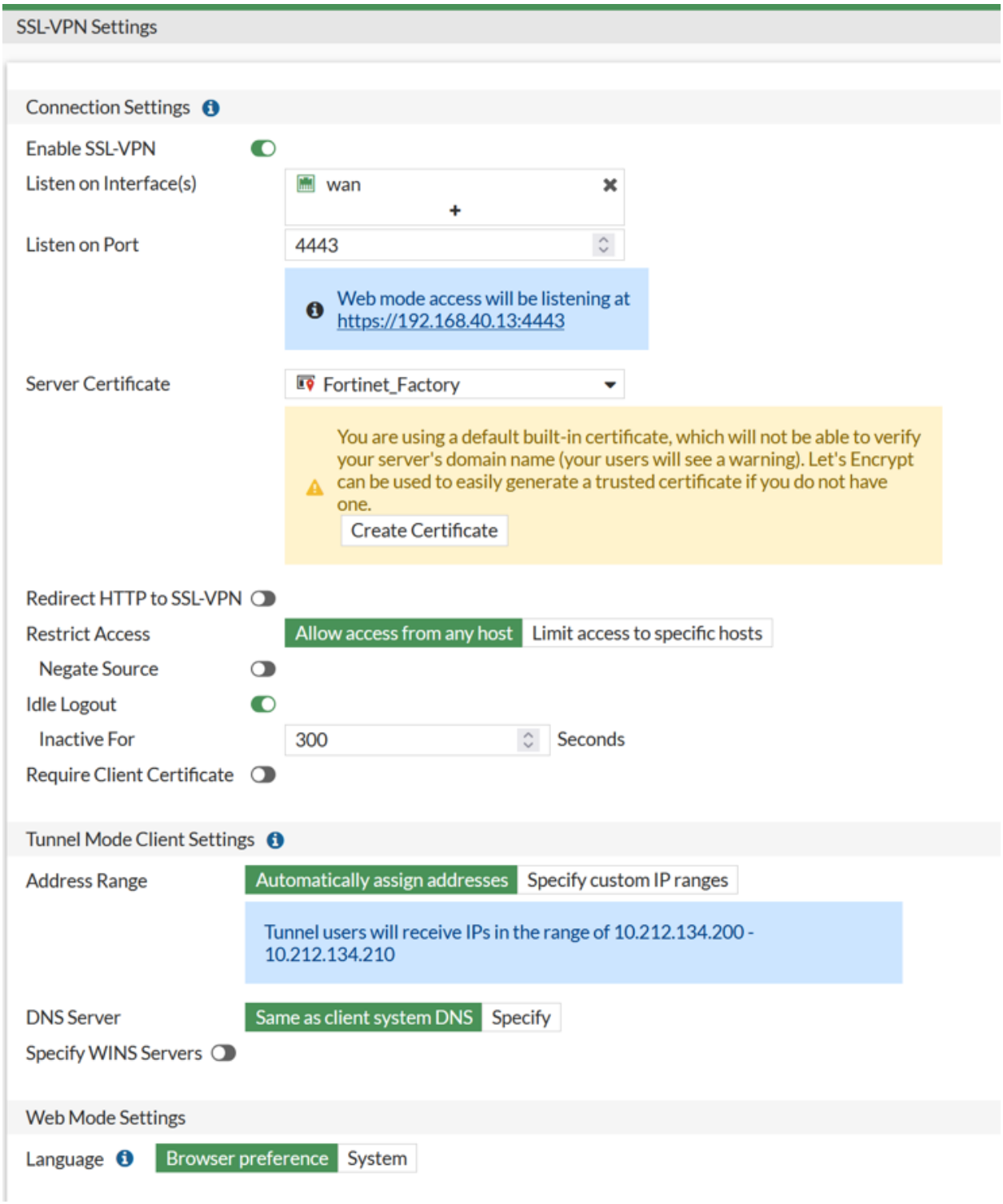
Again, since we are configuring tunnel mode access, this simplifies the configuration steps that we have to complete.

Lab Commands:

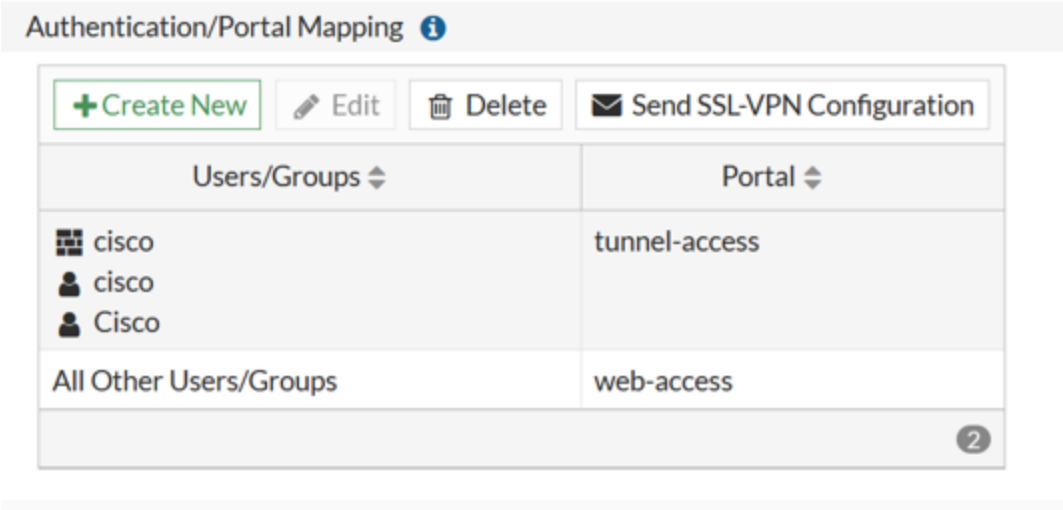
1) First, navigate to the VPN > SSL-VPN settings area. Create a new VPN, and populate it with the following fields:



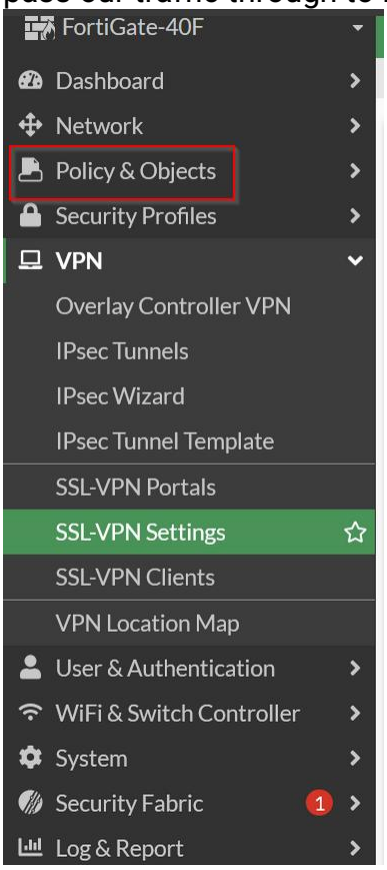
2) Ensure that the listening port is on a non-commonly used port. This is to prevent the VPN tunnel from potentially overlapping with any guest services. The rest of the settings can stay default. We want an Idle Logout for extra security, and our tunnel mode should automatically assign addresses (these addresses will be able to ping in our internal network).



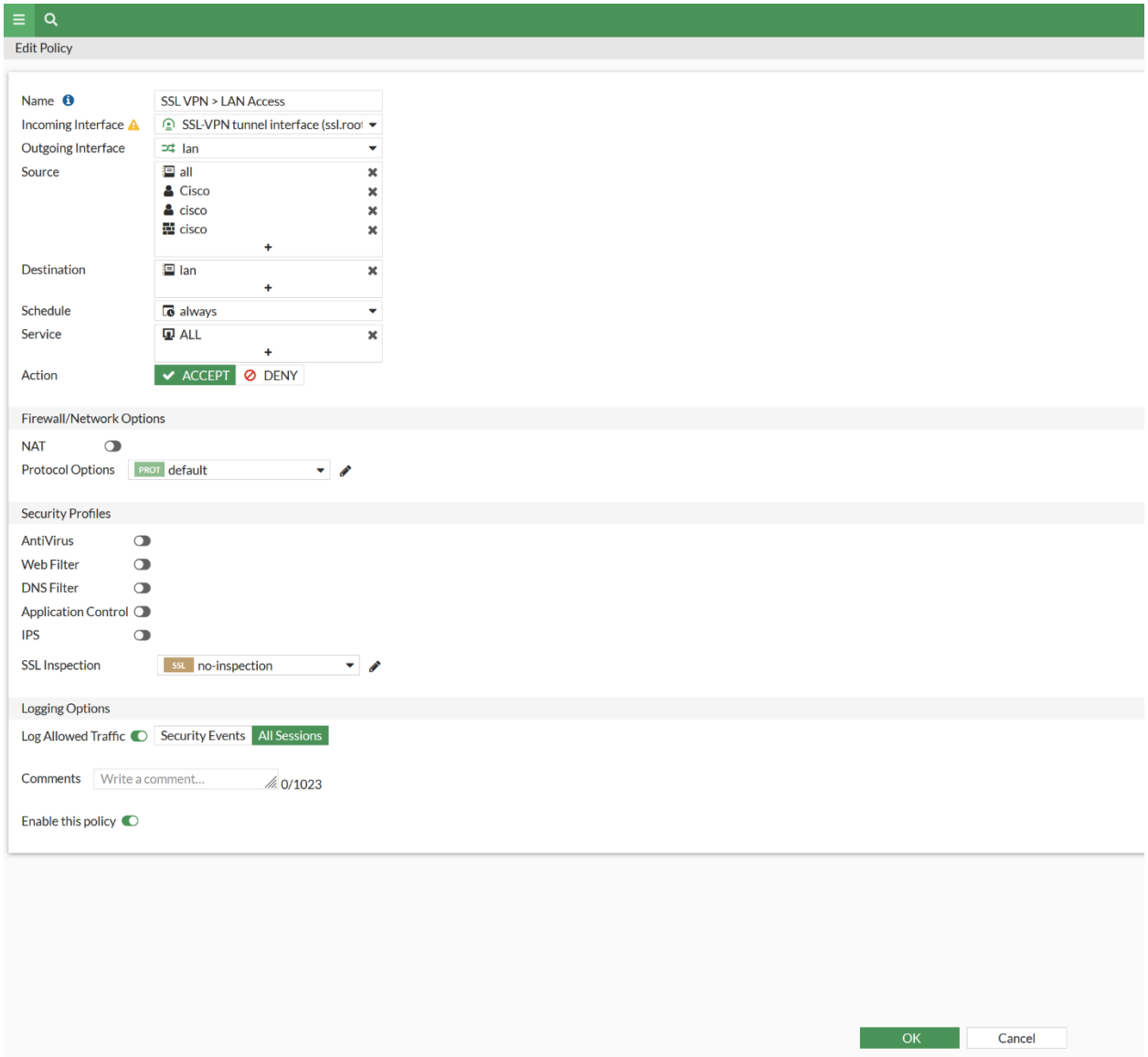
3) Finally, make sure that the user groups are set to allow FortiClient to login with certain credentials. You will need to know these credentials when connecting with FortiClient. In this case, we have a user group allowed.



4) Next, we need to create a firewall policy to allow our SSL VPN traffic to actually pass our traffic through to LAN. Navigate to the Policy and Objects section.



5) Create a new firewall policy. Populate the fields, setting the incoming interface as the SSL VPN object that we created, allowing it to pass to LAN. Allow all traffic to flow through from the previously determined user groups.

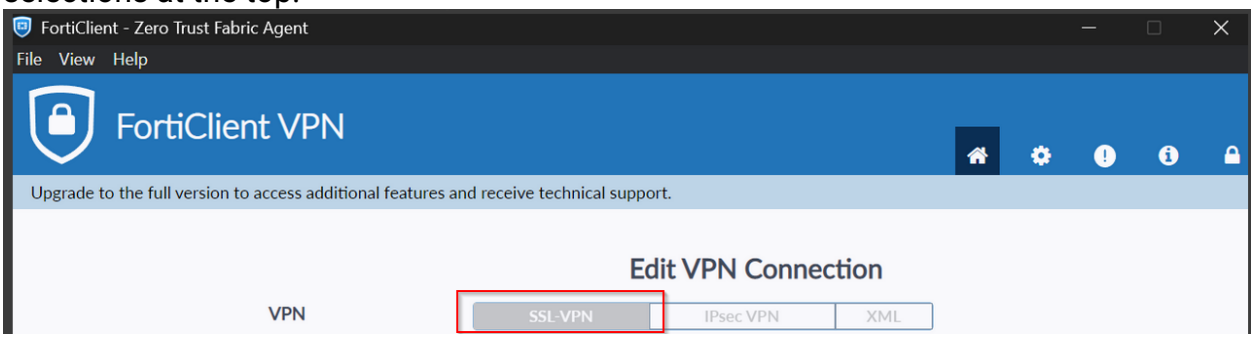


6) After creating this policy, our SSL VPN configuration is completed! Now, we have to test our configuration by connecting via FortiClient and RDP'ing to an internal machine. Navigate to https://www.fortinet.com/support/product-downloads and download the VPN-only file.

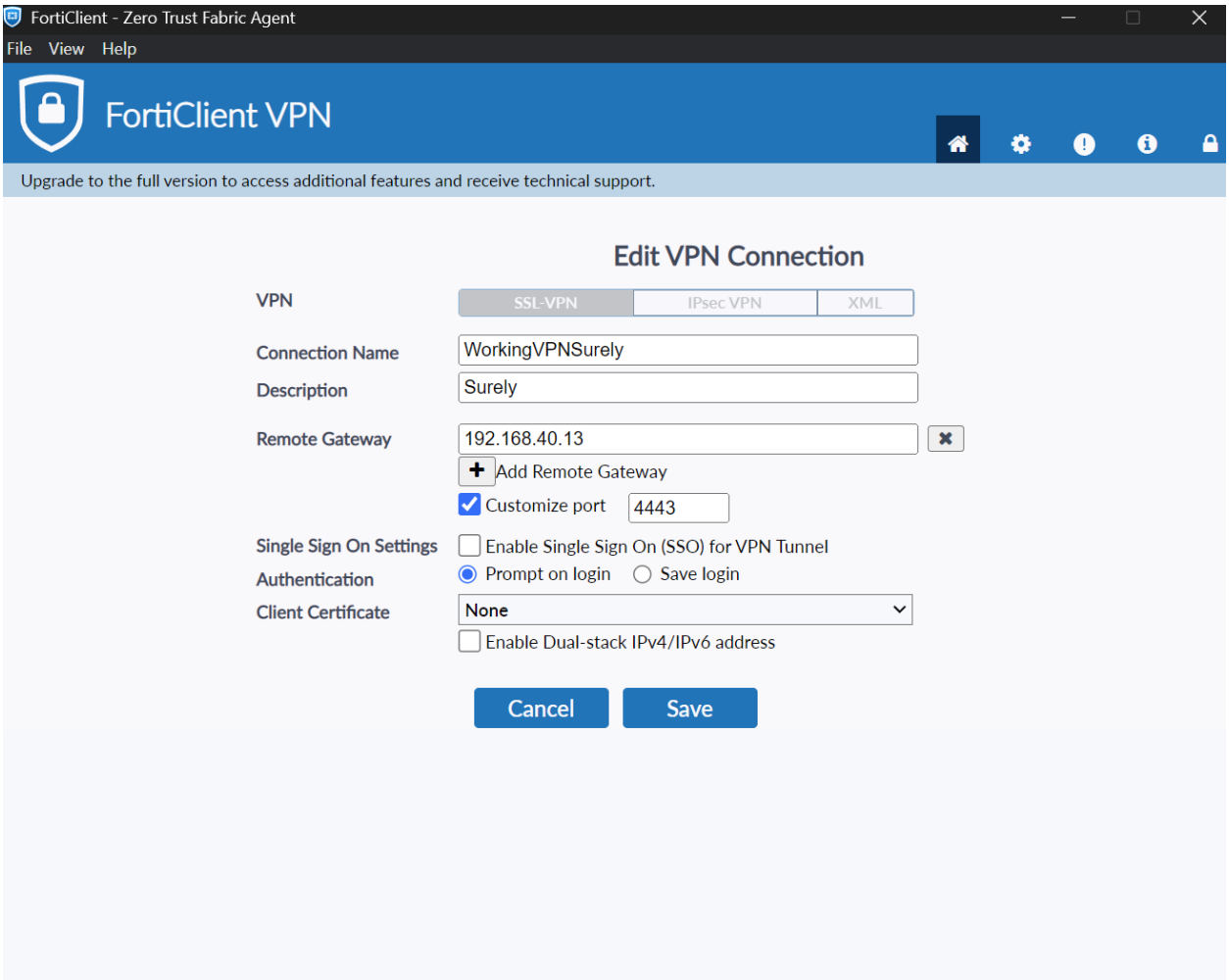
FortiClient VPN-only

Upgrade to one of the Enterprise tiers for Support, central management, zero trust posture assessment, and more.

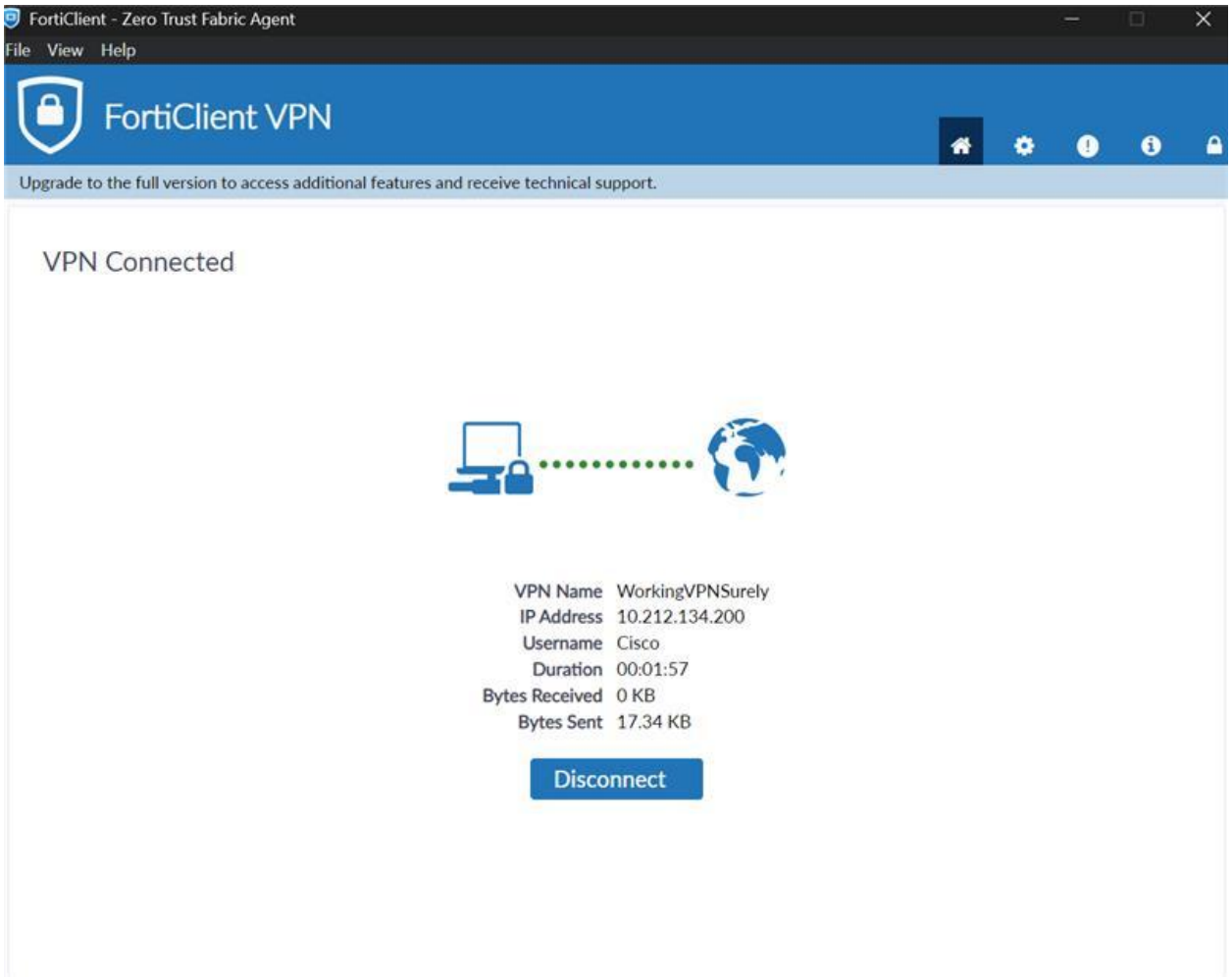
7) Go through the installation, and create a new SSL-VPN tunnel from the three selections at the top.



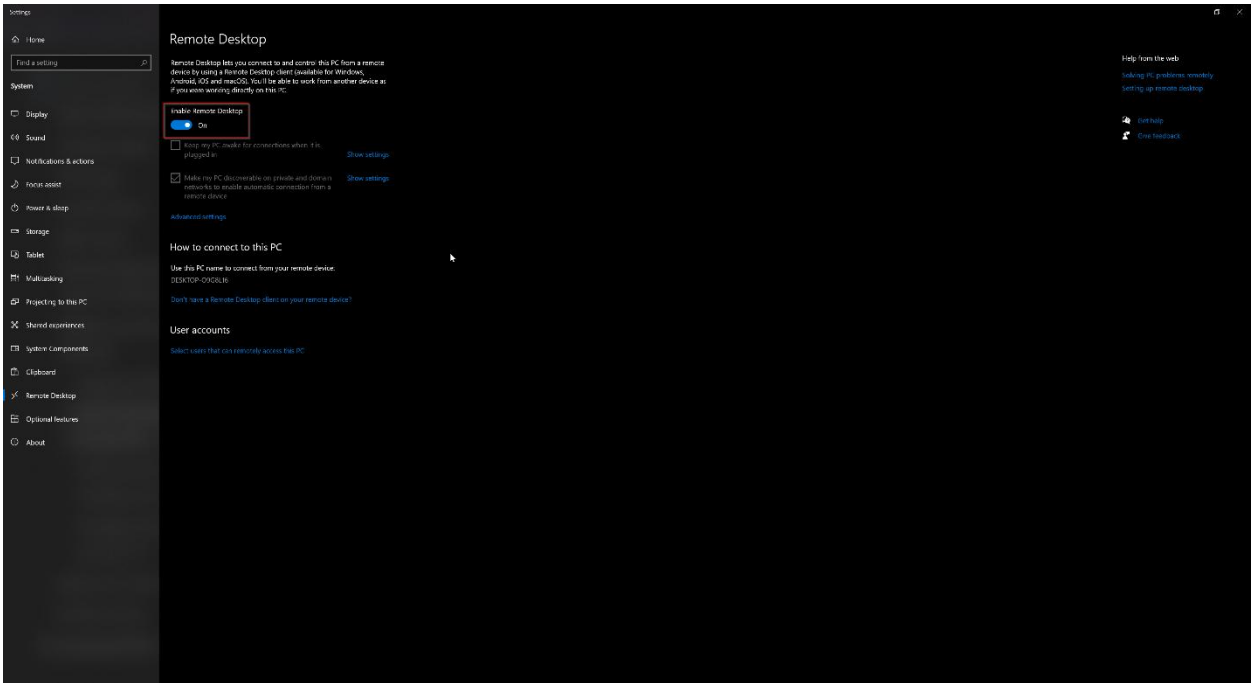
8) Populate the fields with the outwards IP address of the firewall configuration. In this case, our outward facing IP is actually a private network, as this is a simulated environment. However, in a real-world configuration, this global IP should be accessible from the broader internet, and assigned statically by an ISP. Add the target port (4433).



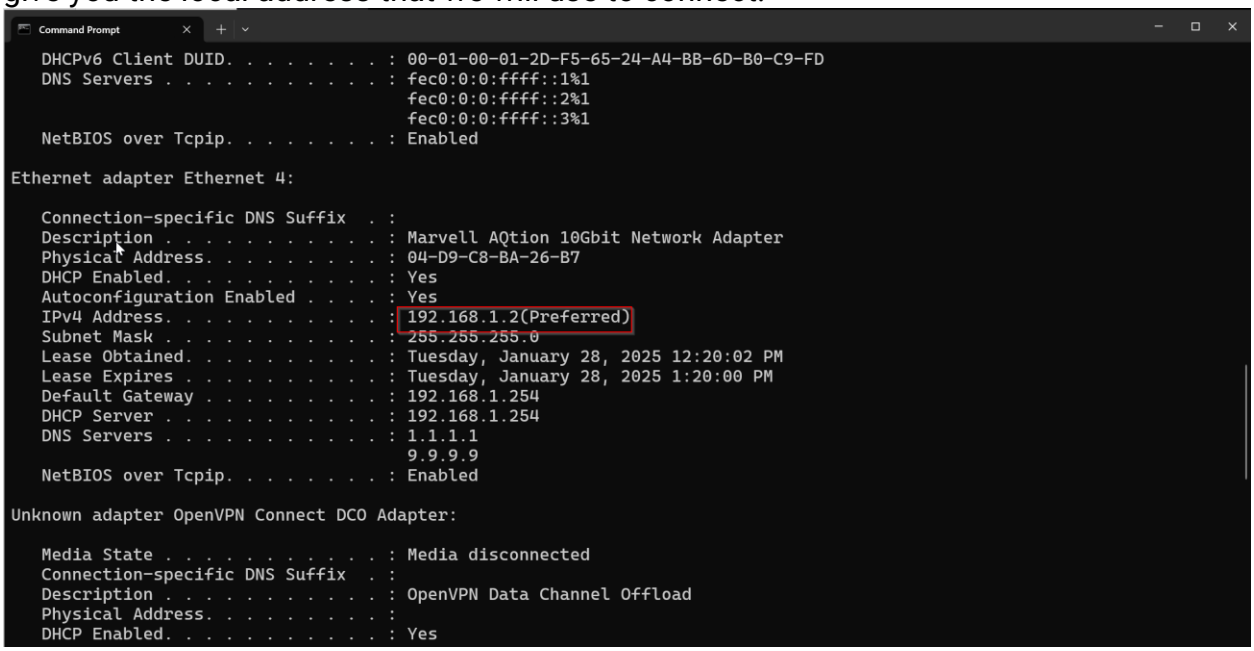
9) Save the VPN and connect to it. Expected behavior should be a loading screen, and then a successful connection to the IP.



10) Ensure that Windows RDP connections are turned on in Windows.

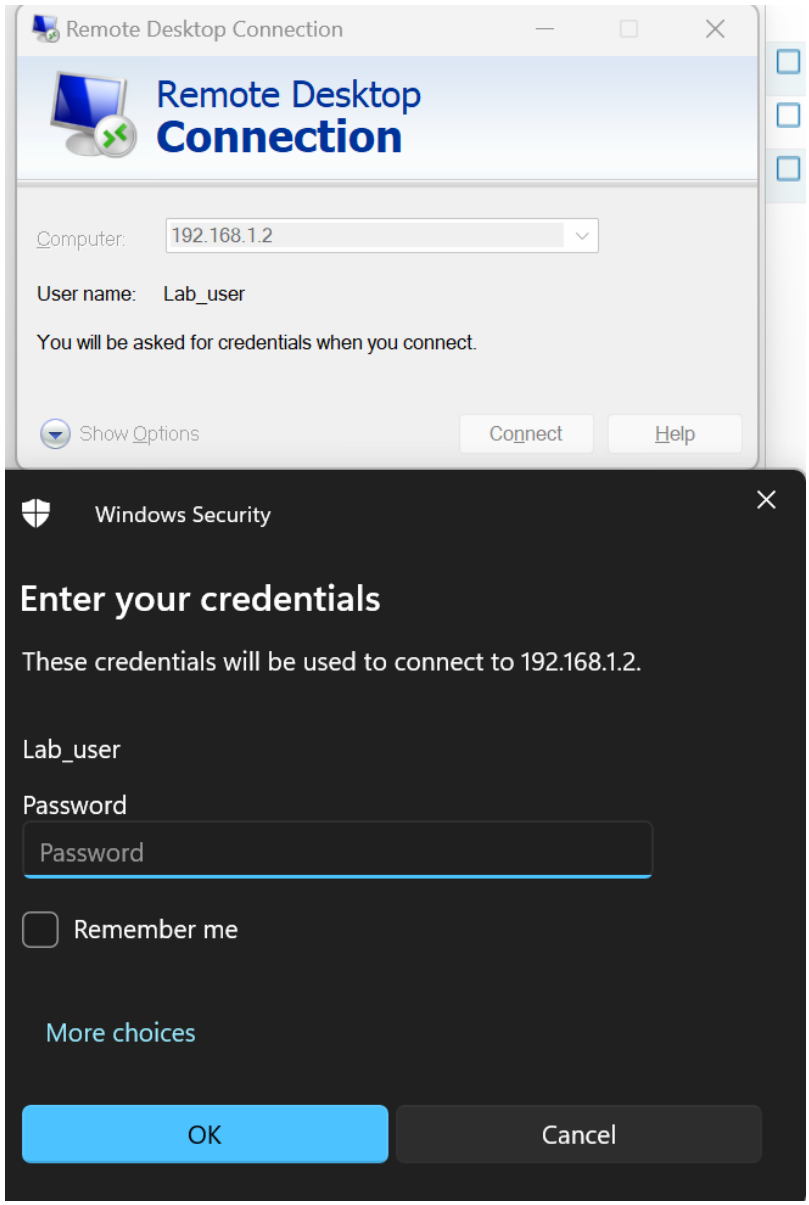


11) Find the IP address of the RDP host using “ipconfig” in a CMD terminal. This will give you the local address that we will use to connect.

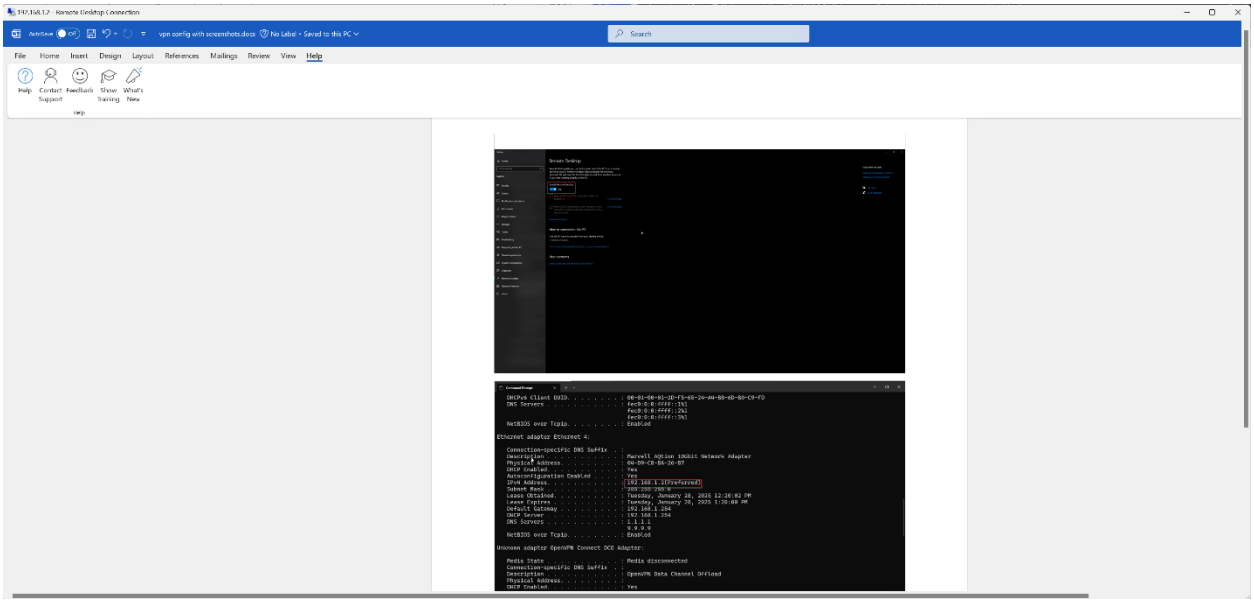


```
Command Prompt

   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2D-F5-65-24-A4-BB-6D-B0-C9-FD
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Marvell AQtion 10Gbit Network Adapter
   Physical Address. . . . . . . . . : 04-D9-C8-BA-26-B7
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.1.2(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Tuesday, January 28, 2025 12:20:02 PM
   Lease Expires . . . . . . . . . . : Tuesday, January 28, 2025 1:20:00 PM
   Default Gateway . . . . . . . . . : 192.168.1.254
   DHCP Server . . . . . . . . . . . : 192.168.1.254
   DNS Servers . . . . . . . . . . . : 1.1.1.1
                                       9.9.9.9
   NetBIOS over Tcpip. . . . . . . . : Enabled

Unknown adapter OpenVPN Connect DCO Adapter:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : OpenVPN Data Channel Offload
   Physical Address. . . . . . . . . :
   DHCP Enabled. . . . . . . . . . . : Yes
```

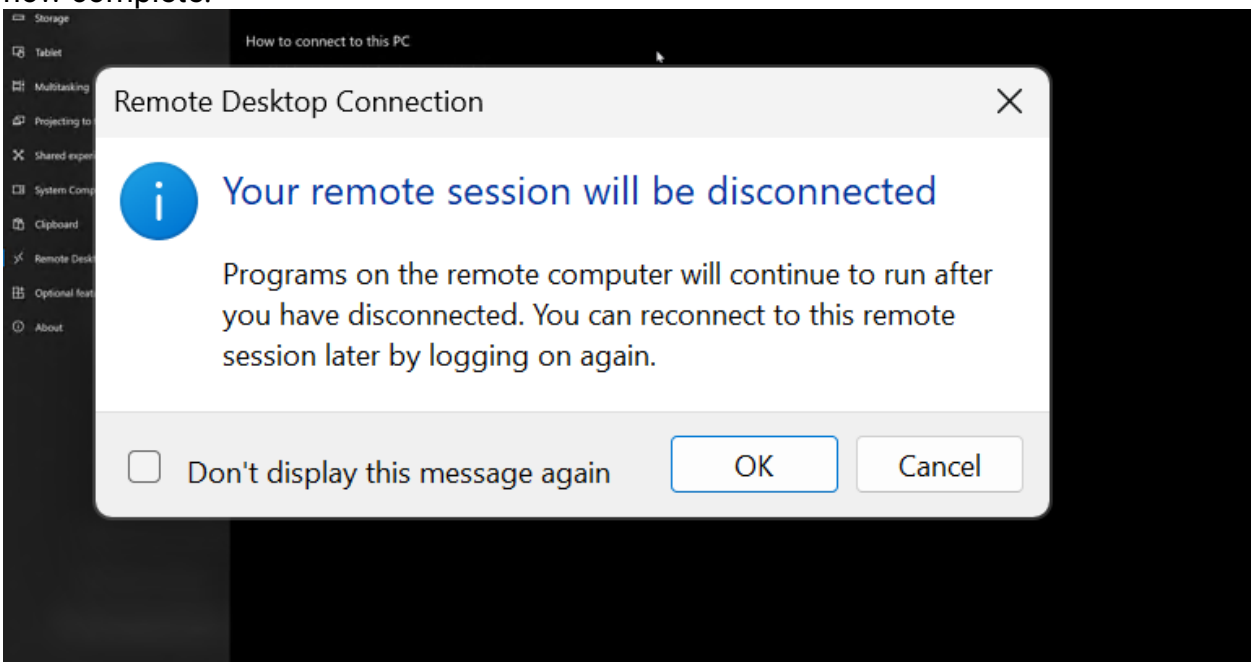
12) On the machine with the VPN connected, initiate an RDP connection.



13) Success! Our VPN connection has initiated a tunnel in which we can access local resources, like a computer through RDP!



14) Disconnect the VPN client and stop the RDP connection. Our configuration is now complete.



Problems:

6) **Required Firewall Policy**
At first, we were unaware that a firewall policy was required for our SSL VPN to pass through to LAN. We assumed that the Fortinet UI would automatically create this firewall policy after we set the SSL VPN object, but it did not. This caused the connection to fail. Due to incorrectly configured logging, this didn't show up either. After adding the policy, everything worked fine.

7) External IP Configuration
   Due to the nature of this lab environment, our “external” facing IP is actually an internal, private network configured on the broader Cisco local network. We forgot about this fact, and were searching for a public IP at first to connect to. The IP we actually should have been using was the one attached to our WAN port, in the 192.168.40.X range.

Conclusion:

Overall, this lab was a great introduction to the SSL VPN capabilities of Fortinet firewalls. It definitely showed that SSL VPN’s are far easier to configure on Fortinet than Palo Alto. The Fortinet firewall GUI takes care of the heavy work, and there is little configuration involved for us.

IPSec Site-to-Site VPN on FortiGate 40F

Ryan Chen

Cybersecurity

Purpose:

The primary purpose of this lab is to configure and test an IPsec VPN on a generic Fortinet Fortigate 40F firewall. Building off of previous information form the SSL VPN configuration, as well as basic networking concepts learned through Palo Alto configuration, this lab aims to successfully setup an IPsec based Site to Site VPN.

Background Information:

A VPN, or virtual private network, is a commonly used service that provides a multitude of different privacy, security, and convenience functions. At its core, they work by encrypting traffic between an end-user and the server, allowing a secure connection to be established from anywhere. Given that this traffic is encrypted, it is essentially improbable for an adversary to read it. Even if they do manage to get hold of the traffic by sniffing it off the network, they will be unable to read it without the key.

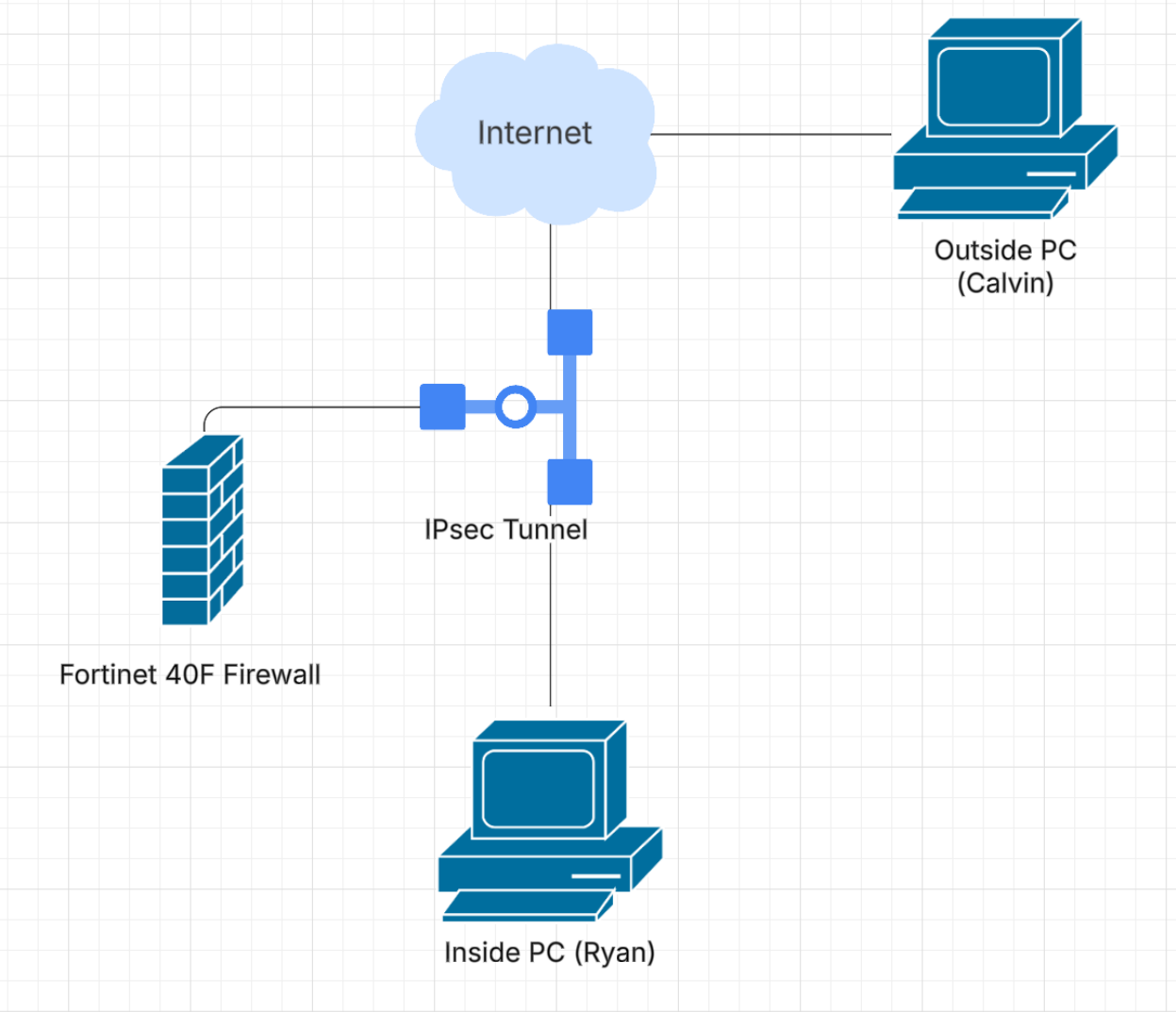
For this reason, VPN's are commonly used to secure traffic from an unsecure network into a secure network. By establishing this secure connection, attacks like a MITM (Man in the Middle) become less of a threat, since traffic is encrypted. Likewise, people also use VPN's for their ability to "tunnel" traffic. This refers to the ability to establish a connection that emulates the device directly on a local network. This is useful for many different use cases, but commonly required by companies when employees are doing remote work. In order to keep company traffic secure even from an unsecure network, companies can configure a VPN to tunnel employee traffic back into the network. This allows employees to work remotely on company networks, all while keeping traffic secure. In our test case, this is the primary usage that we aim to target with our configuration.

Now that we know the basic use cases of a VPN, we can start to cover how it works at a base level. Typically for a VPN to successfully send and receive traffic, end users must connect through an application configured by the VPN provider. This application works by establishing a secure connection through the use of certificates and encryption, allowing end user traffic to be encrypted in transit and decrypted by the VPN at the end.

This lab targets the IPsec protocol based VPN, which is commonly used to connect two different enterprise sites (in different locations) together. This is typically used to provide security in cases where companies may need to remote access into assets from a different location. IPsec is an older protocol, and was the primary VPN configuration protocol before SSL/TLS existed. Configuration is slightly more complicated, but it generally provides more robustness and is a widely used and trusted option for VPNs.

Finally, one of the key concepts of this lab is remote access. For this purpose, we use the RDP (Remote Desktop Protocol) provided by Windows. This protocol allows us to configure one machine inside the network as a server, and the other machine as a end user that will ultimately establish a connection to the server to demonstrate that the VPN is working. RDP gives us the ability to remotely view another Windows desktop, a common use case for companies that want to enable employees to do remote work. We demonstrate functionality of the VPN through the FortiClient connection, and then through RDP.

Network Diagram:



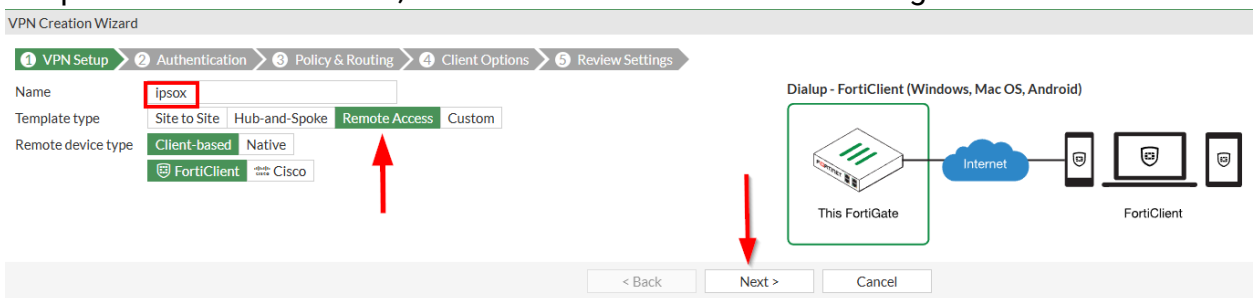
Lab Summary:

IPsec VPN configuration, while still relatively simple, is indeed more complicated than SSL. Fortinet makes it easy to create the IPsec VPN with their built in IPsec VPN Wizard

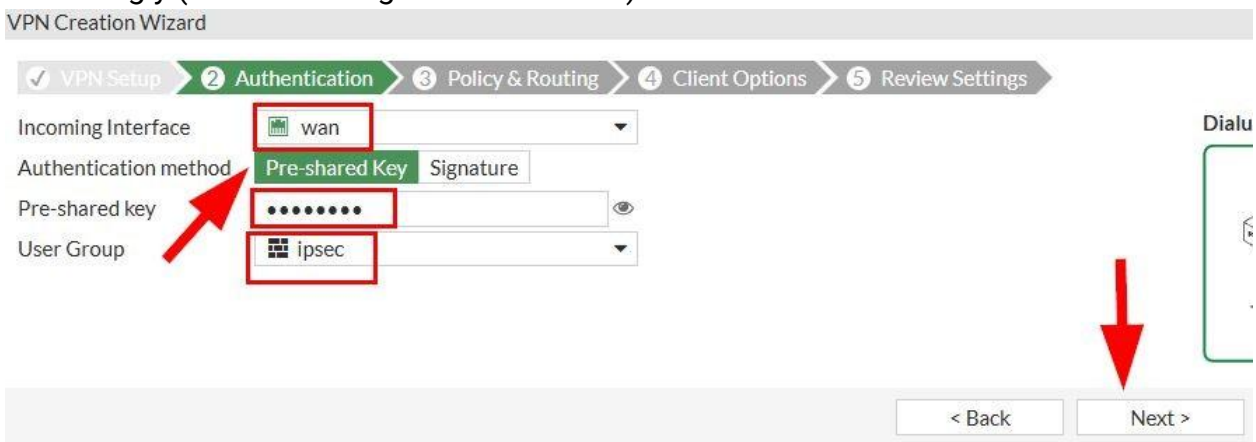
tool. This tool automatically configures most of the things, but custom policies must still be created for both ends of the connection in order to allow traffic to actually pas through. Overall, the process is not difficult, simplifying deployment a lot.

Lab Commands:

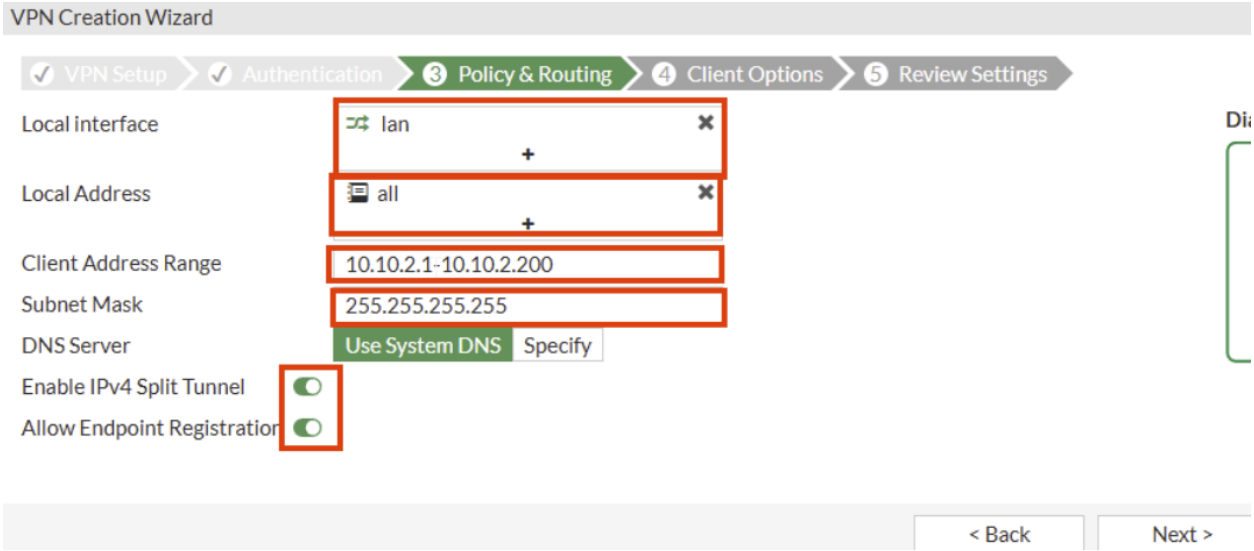
1) Navigate to VPN > VPN Wizard. Create a new VPN and give it a name. Set the template to Remote Access, and make sure that the VPN is using FortiClient.



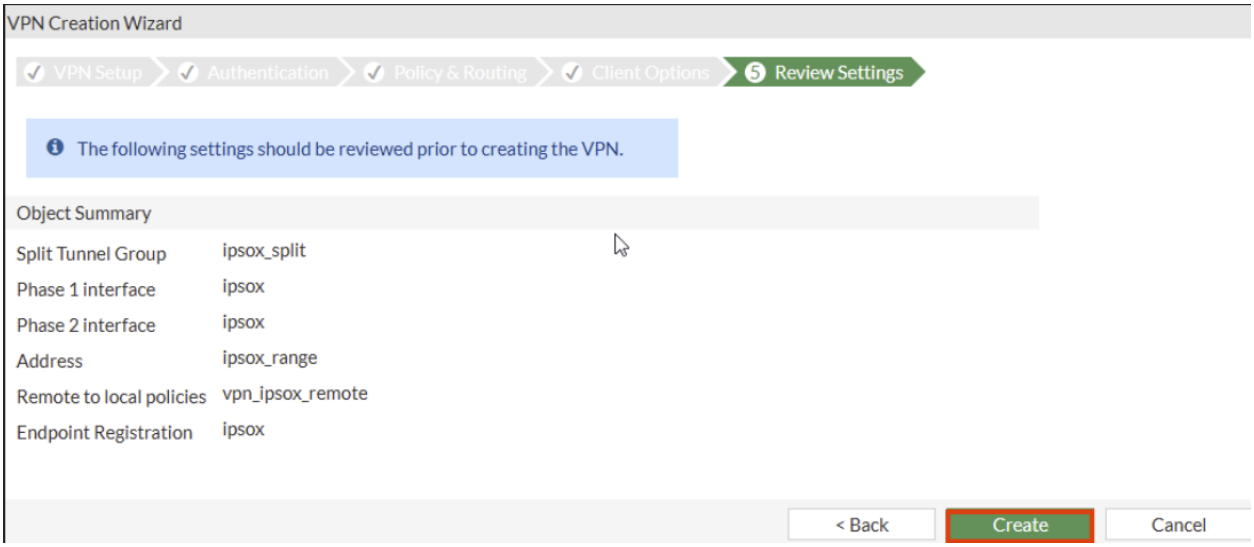
2) Set the agreed upon PSK with the other partner end, and set the User Group accordingly (we later changed ours to Cisco).



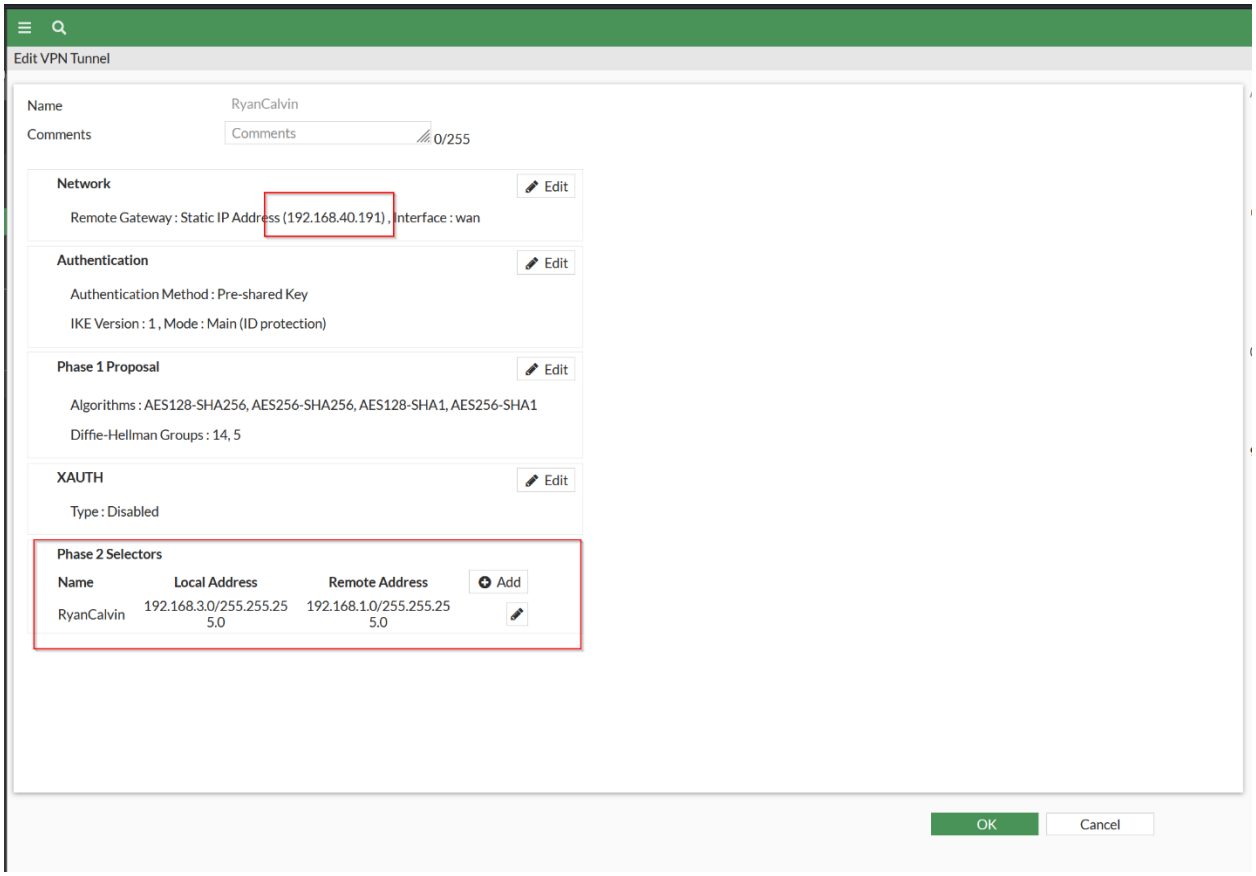
3) Set the address and LAN interface for the receiving end.



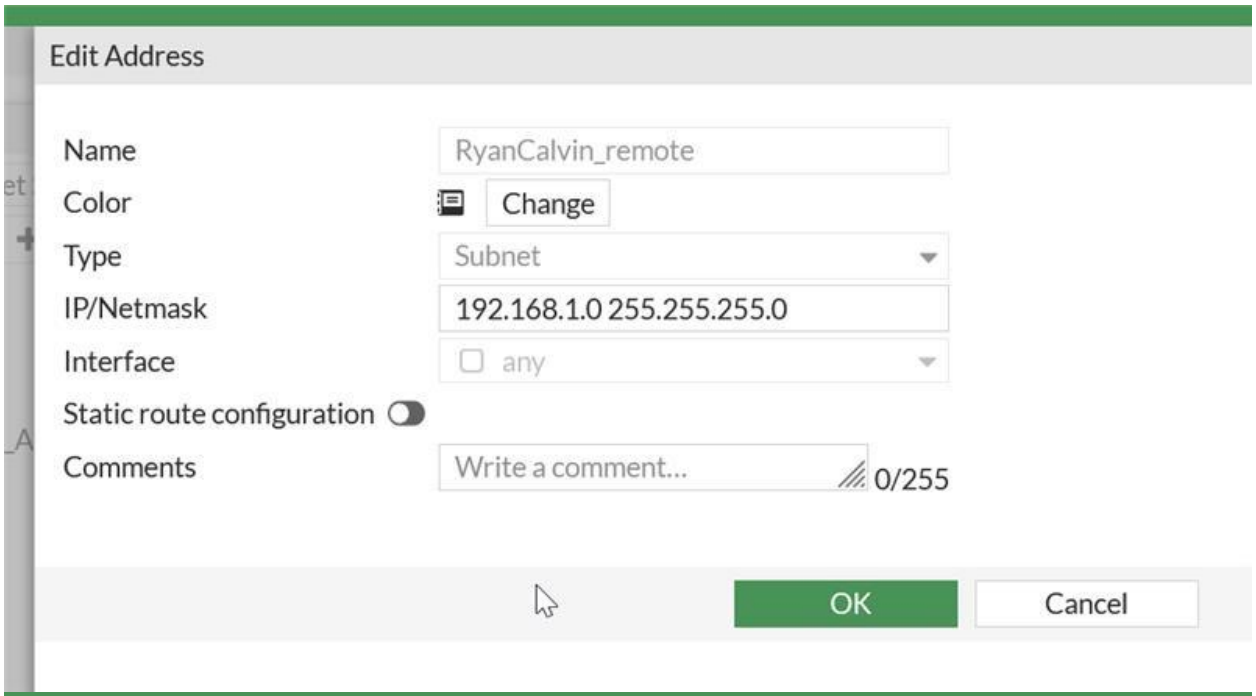
4) Create the new VPN object. The process will automatically create a new VPN tunnel. However, we still need to create the firewall policies to allow traffic to flow.

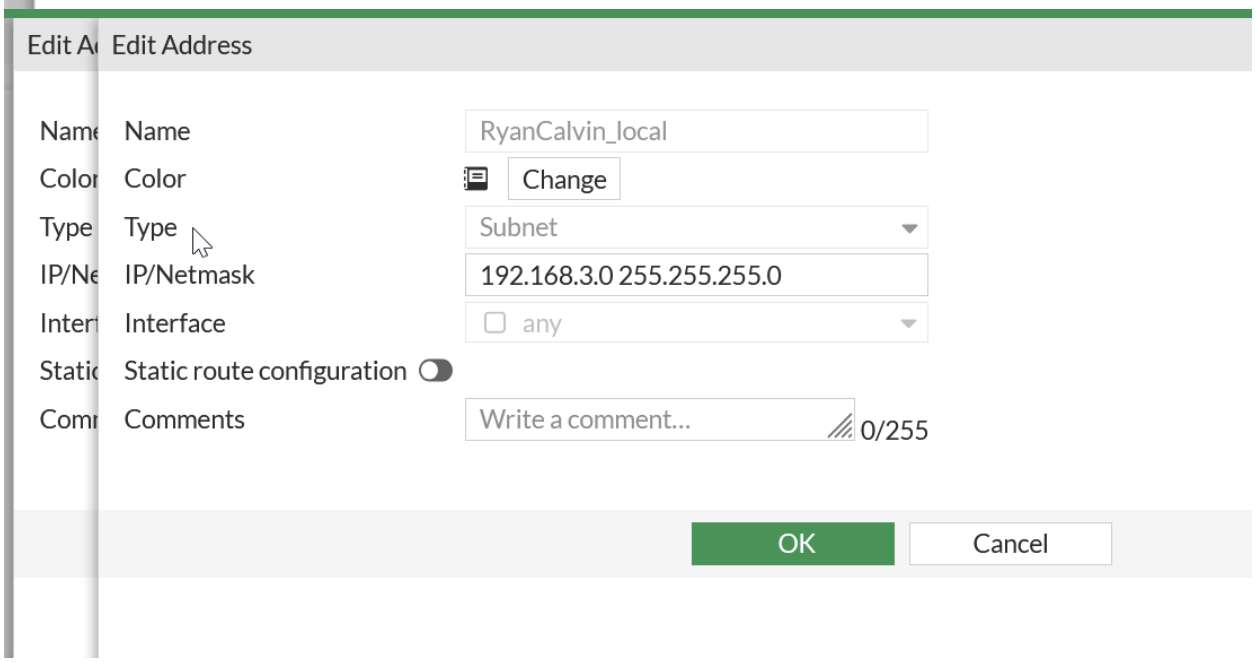


5) Upon creating the VPN tunnel, check that the object has the correct addresses. You may define addresses either with Fortinet objects, or directly type them in. We later changed our configuration to explicitly show address spaces.



6) Before creating firewall policies, navigate to Policies & Objects > Addresses and create a new Address Group. We need to create two different address groups, one for our local address space, and one for the remote address space. We chose our local subnet to be 192.168.3.0, and Calvin, our lab partner, was set as 192.168.1.0.





7) Now, create a new firewall policy. You will need two policies: one to allow traffic from the tunnel to LAN, and one to allow LAN back to the tunnel. You may create them in either order.

Edit Policy

Name	RyanCalvin
Incoming Interface	lan
Outgoing Interface	RyanCalvin
Source	RyanCalvin_local
Destination	RyanCalvin_remote
Schedule	always
Service	ALL
Action	ACCEPT / DENY

Firewall/Network Options

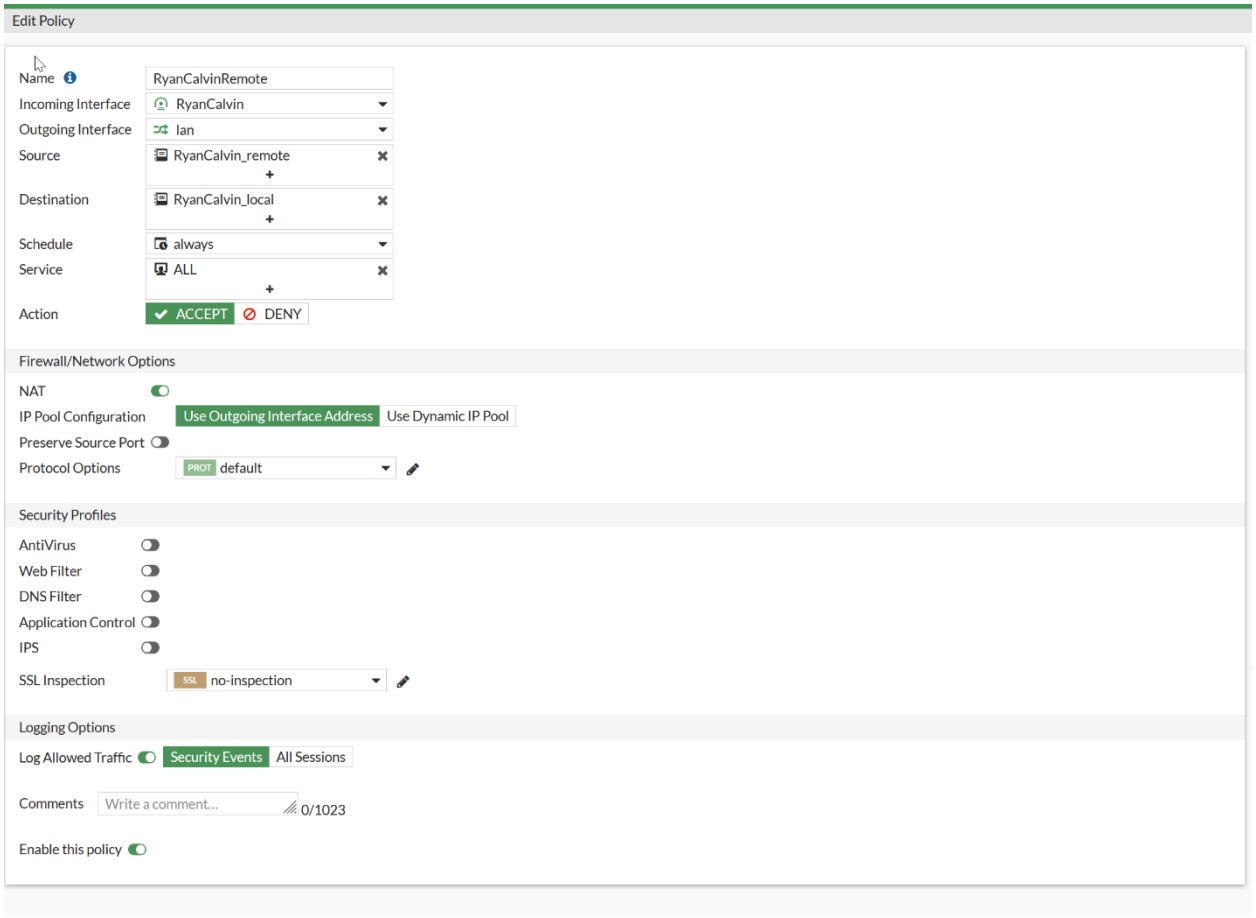
NAT	ON
IP Pool Configuration	Use Outgoing Interface Address / Use Dynamic IP Pool
Preserve Source Port	OFF
Protocol Options	PROT default

Security Profiles

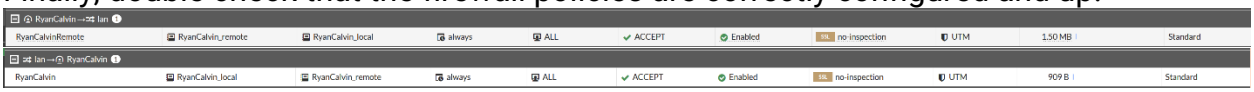
AntiVirus	OFF
Web Filter	OFF
DNS Filter	OFF
Application Control	OFF
IPS	OFF
SSL Inspection	SSL no-inspection

Logging Options

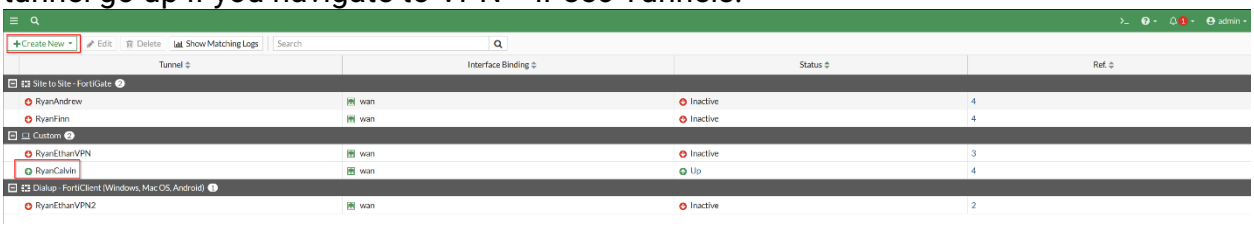
Log Allowed Traffic	ON — Security Events / All Sessions
Comments	Write a comment... 0/1023
Enable this policy	ON



8) Finally, double check that the firewall policies are correctly configured and up.



9) Now that the firewall policies are up, our configuration is complete. If the other end has the same configuration and everything is connected, you should see the tunnel go up if you navigate to VPN > IPsec Tunnels.



10) With the tunnel up, all that's left to do is connect via FortiClient, and then initiate an RDP connection. This was done on our partner's device, and as such, there is no screenshot to display here. You can double check the configuration is working by looking at the firewall policies. You should see that the data fields increase in size, as the amount of data between the two tunnels flows. This means the configuration is successfully completed!

Problems:

8) **Required Firewall Policy**
   At first, we were unaware that a firewall policy was required for the IPsec VPN, similar to the SSL VPN. We also incorrectly assumed that the Fortinet UI would automatically create this firewall policy after we created the IPsec object, but it did not. This caused the connection to fail. Upon adding the policy, everything worked fine.
9) **IPsec Tunnel Down**
   Though the IPsec tunnel may bring itself up automatically if the device on the other end of the tunnel is configured correctly, it also has the possibility not to go up by itself. We were unaware of this the first time, and realized that we had to manually bring the tunnel up. There is an option to Bring Tunnel Up in the IPsec tunnel page, and upon clicking this, the tunnel turns green.
10) **IP Address Ranges**
   Our IP address ranges for one of the IPsec tunnel endpoints was configured with poor subnetting, and as a result cut out the address space that the device was located in. Upon changing this, we were able to initiate an RDP connection, and the configuration worked as expected.

Conclusion:

This lab was definitely more challenging than the SSL VPN, however, it provides a great introduction to IPsec configuration on Fortinet devices. The VPN Wizard makes the process relatively easy, having only to complete the firewall policy parts. This lab was also a great refresher on debugging, since our tunnel was initially not working. We realized later on that it was an issue with the subnet configuration, as we had mistakenly set the subnet to a value smaller than the actual contained addresses. To solve this issue, we were required to work backwards.