COMP702 Scaling Networks Case Study

Josh Campbell

**Topology**

A diagram of a computer network

AI-generated content may be incorrect.

**Network Addressing Scheme Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Site** | **Number of hosts** | **Developers VLAN host required** | **Management VLAN host required** |
| Wellington | 200 | 180 | 20 |
| Auckland | 50 | 40 | 10 |
| Hamilton | 50 | 40 | 10 |
| Wellington to Auckland | 2 |  |  |
| Wellington to Hamilton | 2 |  |  |
| Auckland to Hamilton | 2 |  |  |

**Wellington Site Table**

|  |  |  |
| --- | --- | --- |
| **VLAN** | **IPv4** | **IPv6** |
| Developers (10) | 10.0.1.0/24 (1-254) | 2001:db8:1:1::/64 |
| Management (50) | 10.0.0.0/27 (1-30) | 2001:db8:1:2::/64 |
| Secure\_Down (99) |  |  |
| Native (150) |  |  |

**Auckland Site Table**

|  |  |  |
| --- | --- | --- |
| **VLAN** | **IPv4** | **IPv6** |
| Developers (10) | 10.0.2.0/26 (1-62) | 2001:db8:2:1::/64 |
| Management (50) | 10.0.2.64/28 (65-78) | 2001:db8:2:2::/64 |
| Secure\_Down (99) |  |  |
| Native (150) |  |  |

**Hamilton Site Table**

|  |  |  |
| --- | --- | --- |
| **VLAN** | **IPv4** | **IPv6** |
| Developers (10) | 10.0.3.0/26 (1-62) | 2001:db8:3:1::/64 |
| Management (50) | 10.0.3.64/28 (65-78) | 2001:db8:3:2::/64 |
| Secure\_Down (99) |  |  |
| Native (150) |  |  |

Point to point Table

|  |  |  |
| --- | --- | --- |
| **Name** | **IPv4** | **IPv6** |
| Wellington to Auckland | 10.0.4.0/30 | 2001:db8:4:1::/64 |
| Wellington to Hamilton | 10.0.4.4/30 | 2001:db8:4:2::/64 |
| Auckland to Hamilton | 10.0.4.8/30 | 2001:db8:4:3::/64 |

**Device Interface Addressing Table**

\* Means that DHCP will be used later so it may be different

|  |  |  |
| --- | --- | --- |
| **Device** | **IP address** | **Default Gateway** |
| R1  Serial 0/1/1 | 202.14.63.1 | 202.14.63.3 |
|  |  |
|  |  |
| R1  Gigabit Ethernet 0/0/0.50  VLAN 50 | 10.0.0.1/27 |  |
| 2001:db8:1:2::1/64 |  |
|  |  |
| R1  Gigabit Ethernet 0/0/0.10  VLAN 10 | 10.0.1.1/24 |  |
| 2001:db8:1:1::1/64 |  |
|  |  |
| R1  Serial 0/1/0 | 10.0.4.5/30 |  |
| 2001:db8:4:2::1/64 |  |
|  |  |
| R1  Serial 0/2/0 | 10.0.4.1/30 |  |
| 2001:db8:4:1::1/64 |  |
|  |  |
| R1  Loopback 1 |  |  |
|  |  |
|  |  |
| R1  Loopback 2 |  |  |
|  |  |
|  |  |
| R2  Serial 0/1/1 | 10.0.4.2/30 |  |
| 2001:db8:4:1::2/64 |  |
|  |  |
| R2  Gigabit Ethernet 0/0/0.50  VLAN 50 | 10.0.2.65/28 |  |
| 2001:db8:2:2::1/64 |  |
|  |  |
| R2  Gigabit Ethernet 0/0/0.10  VLAN 10 | 10.0.2.1/26 |  |
| 2001:db8:2:1::1/64 |  |
|  |  |
| R2  Serial 0/1/0 | 10.0.4.9/30 |  |
| 2001:db8:4:3::1/64 |  |
|  |  |
| R2  Loopback 1 |  |  |
|  |  |
|  |  |
| R2  Loopback 2 |  |  |
|  |  |
|  |  |
| R3  Serial 0/1/0 | 10.0.4.6/30 |  |
| 2001:db8:4:2::2/64 |  |
|  |  |
| R3  Serial 0/1/1 | 10.0.4.10/30 |  |
| 2001:db8:4:3::2/64 |  |
|  |  |
| R3  Gigabit Ethernet 0/0/0.50  VLAN 50 | 10.0.3.65/28 |  |
| 2001:db8:3:2::1/64 |  |
|  |  |
| R3  Gigabit Ethernet 0/0/0.10  VLAN 10 | 10.0.3.1/26 |  |
| 2001:db8:3:1::1/64 |  |
|  |  |
| R3  Loopback 1 |  |  |
|  |  |
|  |  |
| R3  Loopback 2 |  |  |
|  |  |
|  |  |
| S1 SVI  VLAN 50 | 10.0.0.3/27 | 10.0.0.1 |
| 2001:db8:1:2::3/64 | 2001:db8:1:2::1/64 |
|  |  |
| S2 SVI  VLAN 50 | 10.0.0.4/27 | 10.0.0.1 |
| 2001:db8:1:2::4/64 | 2001:db8:1:2::1/64 |
|  |  |
| S3 SVI  VLAN 50 | 10.0.2.66/28 | 10.0.2.65 |
| 2001:db8:2:2::2/64 | 2001:db8:2:2::1/64 |
|  |  |
| S4 SVI  VLAN 50 | 10.0.3.66/28 | 10.0.3.65 |
| 2001:db8:3:2::2/64 | 2001:db8:3:2::1/64 |
|  |  |
| PC1  VLAN 10 | 10.0.1.2/24\* | 10.0.1.1 |
| 2001:db8:1:1::2/64\* | 2001:db8:1:1::1 |
|  |  |
| PC2  VLAN 10 | 10.0.1.3/24\* | 10.0.1.1 |
| 2001:db8:1:1::3/64\* | 2001:db8:1:1::1 |
|  |  |
| PC3  VLAN 50 | 10.0.0.2/27 | 10.0.0.1 |
| 2001:db8:1:2::2/64 | 2001:db8:1:2::1 |
|  |  |
| PC4  VLAN 10 | 10.0.2.2/26 | 10.0.2.1 |
| 2001:db8:2:1::2/64 | 2001:db8:2:1::1 |
|  |  |
| PC5  VLAN 10 | 10.0.3.2/26 | 10.0.3.1 |
| 2001:db8:3:1::2/64 | 2001:db8:3:1::1 |
|  |  |
| TFTP Server  VLAN 50 | 10.0.0.10/27 | 10.0.0.1 |
| 2001:db8:1:2::3/64 | 2001:db8:1:2::1 |
|  |  |

Justification:

I created the IP schema using VLSM according to the minimum size requirement for the VLANs of each site. To make things look nicer I had Wellington Site have 0 and 1 as the 3rd octet of the address, 2 be the 3rd octet of the Auckland Site and 3 be the 3rd octet of the Hamilton Site. All of the site-to-site connections have a 4 as the 3rd octet. This can make things a little quicker and easier to understand. Management VLAN is for network devices for remote configuration purposes and administrator end devices.

**Configuration**

Configuring IP Addresses, VLANs and Inter-vlan Routing onto Interfaces:

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| Change Router interface state to be up | The Gigabit Ethernet interfaces on all of the routers and switches are down by default and need to be manually switched on.  Router(config)#interface gigabitEthernet0/0/0  Router(config-if)#no shutdown |
| Create Router Sub-Interface and enable Inter-Vlan routing on interfaces where applicable | Creating the sub-interfaces on the router interfaces that are going to be connected to the rest of the LAN. This is so that Inter-Vlan routing is possible.  Router(config)#interface gigabitEthernet0/0/0.50  Router(config-subif)#encapsulation dot1Q 50 |
| Set Interface IPv4 address and IPv6 address | Ip addresses are essential for network communication. This network uses Ipv4 and Ipv6 for all private Ips.  Router(config-subif)#ip address 10.0.0.1 255.255.255.224  Router(config-subif)#ipv6 address 2001:db8:1:2::1/64 |
| Set Switch ports connecting to router as trunks | Set all of the switch interfaces that are not directly connected to a pc as a trunk so that all traffic can reach the router/default gateway. It is also a good idea to set them to hard-coded trunk ports instead of a dynamic mode so that threat actors cannot swap ports around and perform malicious attacks. This can also mitigate VLAN double-tagging and hopping attacks.  Switch(config)#interface gigabitEthernet0/1  Switch(config-if)#switchport mode trunk |
| Create and name Vlans on Switches | Create all VLANs planned in the network tables. This includes Developers, Management, Secure\_Down and Native.  Switch(config)#vlan 10  Switch(config-vlan)#name Developers |
| Assign VLAN to switch port | Assign appropriate VLANs to the interfaces of all switches. If PC1 is from VLAN 10 and is connected to fastEthernet0/1 of S1, then VLAN 10 will be assigned to that interface.  Switch(config)#interface fastEthernet0/1  Switch(config-if)#switchport access vlan 10 |
| Allow VLANs on Trunk Link and set native VLAN | Allow certain VLAN traffic through the trunk interfaces and set the native VLAN to the trunk links. By doing this, it will help mitigate VLAN attacks. The VLAN Secure\_Down will not be allowed as traffic should not be able to communicate with other devices on the network.  Switch(config)#interface gigabitEthernet0/1  Switch(config-if)#switchport trunk allowed vlan 10,50,150  Switch(config-if)#switchport trunk native vlan 150 |
| Create Loopback Interfaces | Creating loopback interfaces on all three routers for later.  R1(config)#interface loopback 1  R1(config-if)#exit  R1(config)#interface loopback 2 |
| Wan Links | I configure the appropriate IP addresses on the links between routers. I set the clock rate to the default of 64000.  R1(config)#interface serial 0/2/0  R1(config-if)#ip address 10.0.4.1 255.255.255.252  R1(config-if)#ipv6 2001:db8:4:1::1/64  R1(config-if)#no shutdown  R1(config-if)#clock rate 64000 |
| Configure Default Static Route | A default static route must be configured for traffic going to a destination that is unknown to the router. All traffic on R1 will be sent to the IP 202.14.63.3 which is the IP of the router outside the internal network connected on the other end. R2 and R3 must also be told to send unknown traffic to R1 through the Point-to-point links.  R1(config)ip route 0.0.0.0 0.0.0.0 202.14.63.3  R2(config)#ip route 0.0.0.0 0.0.0.0 10.0.4.1  R3(config)#ip route 0.0.0.0 0.0.0.0 10.0.4.5 |
| Enable IPv6 on all routers | A simple command that will enable IPv6 on all three routers.  R1(config)#ipv6 unicast-routing |
| Verification | Show VLAN brief: |

Basic Configuration:

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| Disable DNS Lookup on devices | This will disable the network device from trying to resolve incorrectly typed commands. Having it on and mistyping a command can be annoying.  Router(config)#no ip domain lookup |
| Set Hostname on all network devices | Set hostname of all Switches and Routers so that it is quick as easy to recognise what device you are currently connected to.  Router(config)#hostname R1  R1(config)# |
| Set Encrypted Privileged exec Password | Set a password to secure access to the Privileged exec mode of all network devices. Anytime someone connected through these devices uses the enable command, they will need to know and type this password.  R1(config)#enable secret cisco |
| Set Console Access Password | Set a password to secure Console access to all devices. This means anytime anyone tries to connect through the console port, they will have to know and type this password.  R1(config)#line console 0  R1(config-line)#password class  R1(config-line)#login |
| Encrypt Clear Text Passwords | This will encrypt all plain-test passwords on the device. Since I used the password command on the Console password it would have been plain-text in the running-config. Now it will be encrypted.  R1(config)#service password-encryption |
| Set MOTD Banner | Configure a Message/Banner to show whenever someone attempts to connect to the network device. Not configuring this may allow someone unauthorized who gains access to this device to not get into legal trouble.  R1(config)#banner motd #Authorized access only# |

Configure SSHv2:

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| Set Domain Name | Essential to have SSH since rsa keys are generated using hostname and domain name.  R1(config)#ip domain name joshc.com |
| Generate RSA keys | RSA keys are required for ssh. It uses a public and private key for authentication. These keys are also used to establish and maintain an encrypted connection.  R1(config)#crypto key generate rsa  Modulus: 1024 |
| Create local admin account | Creating an admin account that will be used to login to the device remotely using SSH.  R1(config)#username admin secret class |
| Create VTY lines and enable SSHv2 | Creating vty lines 0 2 so that 3 users can have an SSH connection to a single network device. I also allow only sshv2 because it’s the latest version and is more secure than telnet which is plain-text communication.  R1(config)#line vty 0 2  R1(config-line)#transport input ssh  R1(config-line)#login local  R1(config-line)#exit  R1(config)#ip ssh version 2 |
| Assign IP address to Management VLAN for Switches | Set the IPv4 and IPv6 address for the switch on the management VLAN. This will give an ipv4 address to the switch allowing admins to remotely connect.  S1(config)#interface vlan 50  S1(config-if)#ip address 10.0.0.3 255.255.255.224 |
| Set default gateway for switches | Giving the switches default gateways will allow for SSH connections from outside the LAN since they will be able to communicate with the router.  S1(config)#ip default-gateway 10.0.0.1 |
| Verification | PC1 SSH into R1: |

Switch Port Security, DHCPv4 & DHCPv6:

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| Secure all unused ports | Making sure all switch ports that are currently not planned to be used should be administratively shutdown and on the Secure\_Down VLAN. This is another way of mitigating VLAN attacks.  S1(config)#interface range fastEthernet 0/3 - 9  S1(config-if)#switchport access vlan 99  S1(config-if)#shutdown |
| Hard code Access ports | Switch ports connected to end devices should be hard-coded Access ports to prevent malicious attackers from connecting their own switches or devices. Access ports also need to be hard coded to enable Port Security.  S1(config)#interface range fastEthernet0/1 - 2  S1(config-if-range)#switchport mode access |
| Enable Port Security | Enables port security on an access port of a switch. Without doing this first, port security cannot be configured.  S1(config)#interface range fastEthernet0/1 – 2  S1(config-if-range)#switchport port-security |
| Setting Sticky Mac-addresses | I will be setting a maximum of 1 sticky mac-address to all hard coded access ports. The company does not support BYOD and employees are not allowed to swap around the devices connected to the network. Sticky will allow the first devices mac-address that connects the interface and maximum 1 will allow only that one. I also set the aging time to 60 minutes. *Due to a limitation on packet tracer, I cannot set aging type but I would of set inactivity.* This means that after 60 minutes of no communication between the pc and the switch, the sticky mac-address would be cleared and need to be relearned. The violation mode I choose was restrict. This will disable communication on the port when an unauthorised device attempts to connect. I will also start working when the authorised device connects back without an administrator needing to reenable the interface. Lastly, this will allow a syslog message to be sent to a configured syslog server to notify administrators.  S1(config)#interface range fastEthernet0/1 –  2  S1(config-if-range)#switchport port-security mac-address sticky  S1(config-if-range)#switchport port-security maximum 1  S1(config-if-range)#switchport port-security aging time 60  *S1(config-if-range)#switchport port-security aging type inactivity*  S1(config-if-range)#switchport port-security violation restrict |
| DHCP Snooping | DHCP snooping will be enabled to mitigate DHCP spoofing and starvation attacks. DHCP spoofing can be dangerous because someone can pretend to be the DHCP server and give out incorrect automatic IP configuration. By enabling DHCP spoofing all interfaces become untrusted and all interfaces a DHCP offer would be expected must be configured manually. This includes trunk links.  DHCP starvation could empty the pool of available IPs from the DHCP server and now no more automatic IP configurations could be given out. This is mitigated by limiting the amount of DHCP discover messages can be sent on all untrusted ports. I also enable DHCP spoofing on only VLAN 10 since that is the only VLAN that DHCP will be used on this network.  S1(config)#ip dhcp snooping  S1(config)#interface gigagbitEthernet0/1  S1(config-if)#ip dhcp snooping trust  S1(config-if)#exit  S1(config)#interface range fastEthernet 0/1 – 24  S1(config-if-range)#ip dhcp snooping limit rate 5  S1(config-if-range)#exit  S1(config)#ip dhcp snooping vlan 10 |
| DHCPv4 on R1 for Wellington (Developers VLAN) | DHCPv4 will be enabled on the server in the Wellington Site for the Developers VLAN. This is because this site has around 180 developer devices and configuring static IPs for that many devices can get complicated. I named it DevDHCPv4 to be easily recognisable. DHCP server configuration on packet tracer is a little limited since you can’t exclude specific addresses. I set the starting address and subnet as per the network addressing schema and set the default-routers IP address. I set the DNS server to googles DNS server (8.8.8.8). Since the DHCP server is on a different VLAN to what the pool is giving out, at the default gateway for Developers I will set a helper address to point to the DHCP server.  Pool Name: DEVDHCPv4  Starting Address: 10.0.1.2  Subnet Mask: 255.255.255.0  Default Gateway: 10.0.1.1  DNS-Server: 8.8.8.8  Max Addresses: 253  R1(config)#interface gigabitEthernet0/0/0.10  R1(config-if)#ip helper-address 10.0.0.10 |
| DHCPv6 on R1 for Wellington (Developers VLAN) | Using R1 as a DHCPv6 DHCP address since there is no DHCPv6 relay command in packet tracer. DHCPv6 is used on the Wellington Developers VLAN as well because manually managing a lot of static IPv6 addresses is not recommended.  R1(config)#ipv6 dhcp pool DEVDHCPv6  R1(config-dhcpv6)#address prefix 2001:db8:1:1::/64  R1(config-dhcpv6)#dns-server 2001:4860:4860::8888  R1(config-dhcpv6)#domain-name JoshC.com  R1(config)#interface gigabitEthernet0/0/0.10  R1(config-if)#ipv6 nd managed-config-flag  R1(config-if)#ipv6 dhcp server DEV\_DHCPv6 |

OSPF Configuration: (do this last)

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| Enabling OSPF | OSPF en enabled when the command router ospf <id> is entered. The ID doesn’t have to be the same to establish adjacencies with each other, but I still set them all to be 10. I also went into the routers configuration mode and made R1 advertise its default route to the other routers.  R1(config)#router ospf 10  R2(config)#router ospf 10  R3(config)#router ospf 10  R1(config)#router ospf 10  R1(config-router)#default-information originate |
| Advertise Adjacent Networks | Going into the router configuration mode, we can tell the router what directly connected networks to advertise.  R1(config)router ospf 10  R1(config-router)#network 10.0.1.0 0.0.0.255 area 0  R1(config-router)#network 10.0.0.0 0.0.0.31 area 0  R1(config-router)#network 10.0.4.0 0.0.0.3 area 0  R1(config-router)#network 10.0.4.4 0.0.0.3 area 0  R2(config)router ospf 10  R2(config-router)#network 10.0.2.0 0.0.0.63 area 0  R2(config-router)#network 10.0.2.64 0.0.0.15 area 0  R2(config-router)#network 10.0.4.0 0.0.0.3 area 0  R2(config-router)#network 10.0.4.8 0.0.0.3 area 0  R3(config)router ospf 10  R3(config-router)#network 10.0.3.0 0.0.0.63 area 0  R3(config-router)#network 10.0.3.64 0.0.0.15 area 0  R3(config-router)#network 10.0.4.8 0.0.0.3 area 0  R3(config-router)#network 10.0.4.4 0.0.0.3 area 0 |
| Hello Intervals & Dead Intervals | It is recommended to keep the default settings as they are based on best practices. However, in this network the link between Wellington (R1) and Auckland (R2) is usually sending a lot of crucial traffic. I want this link to failover faster than normal, without being too often that the traffic generated by the hello messages disrupting the network. The default is 10 for hello messages and 40 for dead intervals. I want to configure 5 seconds for the hello messages on this link, and 20 seconds for the dead interval.  R1(config)#interface serial 0/2/0  R1(config-if)#ip ospf hello-interval 5  R1(config-if)#ip ospf dead-interval 20  R2(config)#interface serial 0/1/1  R2(config-if)#ip ospf hello-interval 5  R2(config-if)#ip ospf dead-interval 20 |
| OSPF Priority | We can set interfaces to have certain priorities to control which router becomes the Designated Router and what is the Backup Designated Router. I want R1 to become the DR and R2 to become the BDR. So interfaces on R1 will be set with priority 255. R2 interfaces will be left with default 1. R3 should not become either DB or BDB so should be set to 0.  R1(config)#interface serial 0/2/0  R1(config-if)#ip ospf priority 255  R1(config)#interface serial 0/1/0  R1(config-if)#ip ospf priority 255 |
| Router ID | The Router ID can also be set using different methods where certain methods are prioritized over others. I will be setting the router IDs manually which takes the highest priority. Having router 1s ID be 1.1.1.1, router 2 be 2.2.2.2 and router 3 to be 3.3.3.3 will make it easy to identify which ID belongs to which router.  R1(config)#router ospf 10  R1(config-router)#router-id 1.1.1.1 |
| Reference Bandwidth | I do not need to change or configure the reference bandwidth on any of these because all three routers are using the same link types. I don’t want to have certain paths that are going to be preferred all the time while all three routers are fully operational. |
| Set one loopback interface as a passive interface | On all three routers, loopback interface 1 will be configured to be the passive interface. Passive interfaces are used to prevent ospf hello packets from being sent out so that you don’t form adjacencies on that interface.  R1(config)#router ospf 10  R1(config-router)#passive-interface loopback 1 |
| Verification | Show IP route:  Show IP OSPF neighbor: |
| Bonus: RIPng for IPv6 routing advertisement | To allow IPv6 traffic out of its own physical network and to other sites, I enable RIPng on all three routers and all interfaces with IPv6 enabled.  R1(config)#ipv6 router rip MY-RIP  R1(config)#interface serial 0/2/0  R1(config-if)#ipv6 rip MY-RIP enable  R1(config)#interface serial 0/1/0  R1(config-if)#ipv6 rip MY-RIP enable  R1(config)#interface gigabitEthernet 0/0/0.10  R1(config-if)#ipv6 rip MY-RIP enable  R1(config)#interface gigabitEthernet 0/0/0.50  R1(config-if)#ipv6 rip MY-RIP enable |

IPsec VPN Tunnel: (do this towards the end)

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| Configure ISAKMP Phase 1 | This business uses multiple WAN links to connect each site; however, they need to ensure connection between Wellington and Auckland is further secure. They require a VPN tunnel to ensure all sensitive traffic is encrypted. Using AES for encryption is highly recommended as it is one of the more secure types of encryptions and isn’t too processor intensive. Group 5 is used to share the key to establish the secure connection before when it was an unsecure connection. While a lot of these settings are considered low in security, this link between sites isn’t using the internet. If they were to be connected over the internet, a lot more processor intensive settings would have been selected. Group 5 is also the highest option available on packet tracer.  R1(config)#crypto isakmp policy 10  R1(config-isakmp)#encryption aes 256  R1(config-isakmp)#hash sha  R1(config-isakmp)#authentication pre-share  R1(config-isakmp)#group 5  R1(config-isakmp)#exit  R1(config)#crypto isakmp key vpn address 10.0.4.2  R2(config)#crypto isakmp policy 10  R2(config-isakmp)#encryption aes 256  R2(config-isakmp)#hash sha  R2(config-isakmp)#authentication pre-share  R2(config-isakmp)#group 5  R2(config-isakmp)#exit  R2(config)#crypto isakmp key vpn address 10.0.4.1 |
| Configure ISAKMP Phase 2 | The protocols used by this IPSec tunnel will be defined here. AES will be used for Confidentiality via encryption. AES is both strong and fast without being so strong that it slows down the network. SHA will be used for integrity as it is the strongest option available and is still considerably fast. I have to create an access-list that allows all traffic between Wellington and Auckland to pass through.  R1(config)#crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac  R1(config)#ip access-list extended VPN-ACL  R1(config-ext-nacl)#permit ip 10.0.0.0 0.0.0.31 10.0.2.0 0.0.0.63  R1(config-ext-nacl)#permit ip 10.0.0.0 0.0.0.31 10.0.2.64 0.0.0.15  R1(config-ext-nacl)#permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.63  R1(config-ext-nacl)#permit ip 10.0.1.0 0.0.0.255 10.0.2.64 0.0.0.15  R1(config-ext-nacl)#exit  R1(config)#crypto map VPN-MAP 10 ipsec-isakmp  R1(config-crypto-map)#set peer 10.0.4.2  R1(config-crypto-map)#set transform-set VPN-SET  R1(config-crypto-map)#match address VPN-ACL  R1(config-crypto-map)#exit  R1(config)#interface serial0/2/0  R1(config-if)#crypto map VPN-MAP  R2(config)#crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac  R2(config)#ip access-list extended VPN-ACL  R2(config-ext-nacl)#permit ip 10.0.2.0 0.0.0.63 10.0.0.0 0.0.0.31  R2(config-ext-nacl)#permit ip 10.0.2.0 0.0.0.63 10.0.1.0 0.0.0.255  R2(config-ext-nacl)#permit ip 10.0.2.64 0.0.0.15 10.0.0.0 0.0.0.31  R2(config-ext-nacl)#permit ip 10.0.2.64 0.0.0.15 10.0.1.0 0.0.0.255  R2(config-ext-nacl)#exit  R2(config)#crypto map VPN-MAP 10 ipsec-isakmp  R2(config-crypto-map)#set peer 10.0.4.1  R2(config-crypto-map)#set transform-set VPN-SET  R2(config-crypto-map)#match address VPN-ACL  R2(config-crypto-map)#exit  R2(config)#interface serial0/1/1  R2(config-if)#crypto map VPN-MAP |
| Verification | Show crypto isakmp sa:    Show crypto ipsec sa: |

NAT & PAT:

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| Configure Outside interface | Configure the interface on R1 that is connected to the ISP. It will have the public IP address of 202.14.63.1 and must be manually turned on.  R1(config)#interface serial0/1/1  R1(config-if)#ip address 202.14.63.1 255.255.255.0  R1(config-if)#no shutdown |
| Set NAT Inside and Outside Interfaces | NAT will be configured on R1 with inside interfaces being: GigabitEthernet0/0/0, Serial0/2/0, Serial 0/1/0 and Outside Interface as: Serial0/1/1. The outside interface is the port that is facing the Internet and has a public IP address.  R1(config)#interface serial 0/1/1  R1(config-if)#ip nat outside  R1(config)#interface gigabitEthernet 0/0/0.10  R1(config-if)#ip nat inside  R1(config)#interface gigabitEthernet 0/0/0.50  R1(config-if)#ip nat inside  R1(config)#interface serial 0/1/0  R1(config-if)#ip nat inside  R1(config)#interface serial 0/2/0  R1(config-if)#ip nat inside |
| PAT ACL | I must create an access list that allows all traffic from my network except for the Hamilton Branch Developers VLAN to go through the outside interface. This is because the company wants to use Dynamic Nat for the developers in the Hamilton Branch.  R1(config)#access-list 10 permit 10.0.0.0 0.0.0.31  R1(config)#access-list 10 permit 10.0.1.0 0.0.0.255  R1(config)#access-list 10 permit 10.0.2.0 0.0.0.63  R1(config)#access-list 10 permit 10.0.2.64 0.0.0.15  R1(config)#access-list 10 permit 10.0.3.64 0.0.0.15 |
| Apply PAT | This one command will enable PAT using the access-list I created earlier and apply it to the outside interface.  R1(config)#ip nat inside source list 10 interface serial 0/1/1 overload |
| Dynamic NAT for Hamilton Branch | The Hamilton Branch does not need to use the internet often, so Dynamic NAT will be used with a pool of 5 public IP addresses. This means that only 5 devices from the Hamilton branch can access the internet at a time.  R1(config)#ip nat pool HAMNAT 202.1.5.1 202.1.5.5 netmask 255.255.255.248  R1(config)#access-list 20 permit 10.0.3.0 0.0.0.63  R1(config)#ip nat inside source list 20 pool HAMNAT |
| Verification | Show IP NAT translations:    Internet access:  The server on the other side of the Internet router has static NAT configured with the web-server role. Here PC1 uses the IP of that webserver to view the example page. |

ACLs:

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| Extended ACL to block SSH to default gateway of Auckland Site. | An extended Access Control list will be used to block SSH connections to the router for the Auckland Developers VLAN. This is because the Auckland Branch has their devices placed in a less secure environment. This will reduce the likelihood of someone unauthorized attempting to ssh into the router. The ACL will be placed as close to the source as possible since it is an extended ACL and will be checking packets coming to the router inwards.  R2(config)#ip access-list extended BlockSSH  R2(config-ext-nacl)#deny tcp 10.0.2.0 0.0.0.63 host 10.0.2.1 eq 22  R2(config)#interface gigabitEthernet0/0/0.10  R2(config-if)#ip access-group BlockSSH in |
| Extended ACL to block HTTPS traffic to the internet on Hamilton Site | On another Extended ACL I add a rule that blocks https traffic from the Hamilton branch going toward the internet. This is because the Hamilton branch employees don’t require web access to perform their tasks and have been getting distracted on the internet. However, before the statement that blocks https to the internet, I put a rule to allow https to the Wellington management network. This is because the company still wants employees to access the HTTPS server located there. The ACL is placed close to the source because it is an extended ACL and this will stop traffic from going deep through the network, only to get denied, wasting bandwidth.  R3(config)#ip access-list extended BlockInternetHTTPS  R3(config-ext-nacl)#permit tcp 10.0.3.0 0.0.0.63 10.0.0.0 0.0.0.31 eq 443  R3(config-ext-nacl)#deny tcp 10.0.3.0 0.0.0.63 any eq 443  R3(config)#interface gigabitEthernet 0/0/0.10  R3(config-if)#ip access-group BlockInternetHTTPS in |
| Allow all other traffic | On the ACLs created earlier I will place an allow any to remove the deny any at the very end. Without this at the end, these ACLs would just deny all traffic.  R2(config)#ip access-list extended BlockSSH  R2(config-ext-nacl)#permit ip any any  R3(config)#ip access-list extended BlockInternetHTTPS  R3(config-ext-nacl)#permit ip any any |
| Verification | BlockHTTPS:    PC5 is a part of the Hamilton Branch and cannot view the https webpage from the internet server. Remember to add the s at the end of https.  PC5 still has access to the internal webserver.    BlockSSH:    Here PC4 attempts to SSH into its default gateway, but it does not reach it. When attempting to ssh into the default gateway of another site (Wellington in this case), It is able to. |

TFTP Server for backups:

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| Copy the running config to the TFTP Server in the Wellington Site. | All three routers running-config files will be saved to the TFTP server with an address of 10.0.0.10. This can be useful in the unfortunate event something goes wrong with a router. The config file scan be used to recover the previous configuration to restore network operations.  R1#copy running-config tftp:  … 10.0.0.10 |
| Verification | Here you can see the config files are stored in the TFTP server. |

CDP & LLDP:

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| CDP | CDP will be enabled globally so that devices can learn information about neighbouring cisco devices. It is normally enabled by default on devices, but to make sure the command showed below was used.  R1(config)#cdp run |
| LLDP | LLDP is an open standard version of CDP, so it will work with non-cisco devices. Unlike CDP, it is not enabled by default so it will be enabled using these commands.  (Due to a limitation on packet tracer, you can not use the bottom two commands. For some reason the routers are not learning each other and I believe that is because I can’t configure the Site-to-Site links with LLDP)  R1(config)#lldp run  *R1(config)#interface serial 0/2/0*  *R1(config-if)#lldp transmit*  *R1(config-if)#lldp receive* |
| Verification | CDP:    LLDP: |

NTP:

|  |  |
| --- | --- |
| **Task** | **Commands Used/Justification** |
| Set R1 as NTP Server Node | R1 will have its time set to the current date and time. Then R1 will be set as ntp master 1, meaning it will be the highest authoritative Node since lower is better.  R1#clock set 12:00:00 14 JUNE 2025  R1#conf t  R1(config)#ntp master 1 |
| Set R2 and R3 as NTP Clients | Using the IP addresses that R2 and R3 use to connect to R1, I set them to receive NTP settings.  R2(config)#ntp server 10.0.4.1  R3(config)#ntp server 10.0.4.5 |
| Verification | R2:    R3: |

Troubleshooting:

|  |  |
| --- | --- |
| **Issue** | **Solution** |
| DHCPv4 on R1 was not working. | Instead of configuring it on the router, I decided to add the role to the TFTP server. On R1, I would have to add a helper address so that the DHCP discover message can reach the server located on the management VLAN. |
| When testing if HTTPS was blocked by the ACL, PC5 could still reach the webpage. | Checking the Internet Server, I saw that I had HTTP and HTTPS enabled, so I disabled HTTP and now the page was getting blocked. |
| Wellington was the only device that could reach the internet. The other two sites would get destination unreachable. | When I saw that message, I realized that the router was not able to reach outside of what networks it knew. Thanks to OSPF it could reach my other sites and not give errors. So that means that I forgot to configure a default static route on R2 and R3 to go to R1. I added them and internet was accessible. |
| IPv6 pings couldn’t get outside of their own physical network. They could still ping devices on different VLANs on the same site. | Researching what could fix this, I found out about RIPng which is similar to OSPF but for IPv6. Enabling this on the links between sites I still ran into issues but then quickly figured out I also have to enable it on the inside interfaces. Then I could ping between sites. |

Testing Connectivity using Pings:

I will ping all end devices and the internet router/internet server from PC1. If PC1 can ping a pc in another site, then PC1 is able to ping all the interfaces of the routers the ping travels through. This is done so that the table doesn’t get overwhelmingly long with 6 end devices pinging each sub-interface of each router. Also PC1 and PC2 may have different addresses later because of DHCP.

|  |  |  |  |
| --- | --- | --- | --- |
| **From** | **To** | **Protocol** | **Screenshot** |
| PC1 | PC2 | IPv4 |  |
| IPv6 |  |
| PC1 | PC3 | IPv4 |  |
| IPv6 |  |
| PC1 | PC4 | IPv4 |  |
| IPv6 |  |
| PC1 | PC5 | IPv4 |  |
| IPv6 |  |
| PC1 | Internet | IPv4 |  |
|  |  |
| PC1 | Server | IPv4 |  |
| IPv6 |  |
| PC2 | PC1 | IPv4 |  |
| IPv6 |  |
| PC2 | PC3 | IPv4 |  |
| IPv6 |  |
| PC2 | PC4 | IPv4 |  |
| IPv6 |  |
| PC2 | PC5 | IPv4 |  |
| IPv6 |  |
| PC2 | Internet | IPv4 |  |
|  |  |
| PC2 | Server | IPv4 |  |
| IPv6 |  |
| PC3 | PC1 | IPv4 |  |
| IPv6 |  |
| PC3 | PC2 | IPv4 |  |
| IPv6 |  |
| PC3 | PC4 | IPv4 |  |
| IPv6 |  |
| PC3 | PC5 | IPv4 |  |
| IPv6 |  |
| PC3 | Internet | IPv4 |  |
|  |  |
| PC3 | Server | IPv4 |  |
| IPv6 | Fail |
| PC4 | PC1 | IPv4 |  |
| IPv6 |  |
| PC4 | PC2 | IPv4 |  |
| IPv6 |  |
| PC4 | PC3 | IPv4 |  |
| IPv6 |  |
| PC4 | PC5 | IPv4 |  |
| IPv6 |  |
| PC4 | Internet | IPv4 |  |
|  |  |
| PC4 | Server | IPv4 |  |
| IPv6 |  |
| PC5 | PC1 | IPv4 |  |
| IPv6 |  |
| PC5 | PC2 | IPv4 |  |
| IPv6 |  |
| PC5 | PC3 | IPv4 |  |
| IPv6 |  |
| PC5 | PC4 | IPv4 |  |
| IPv6 |  |
| PC5 | Internet | IPv4 |  |
|  |  |
| PC5 | Server | IPv4 |  |
| IPv6 |  |