Part 1

Introduction:

"The ISP Head Quarter Manager (IHQM) has consulted us as professional network administrators to implement and secure the communication over the multiaccess network at the ISP Head Quarter. In this report, we will outline the key security measures that have been put in place to ensure the confidentiality, integrity, and availability of the network, including the use of the OSPF routing protocol, switch port security, and additional security measures such as TACACS+ and packet encryption. We will also provide an overview of the network infrastructure and discuss best practices for maintaining the security of the network."

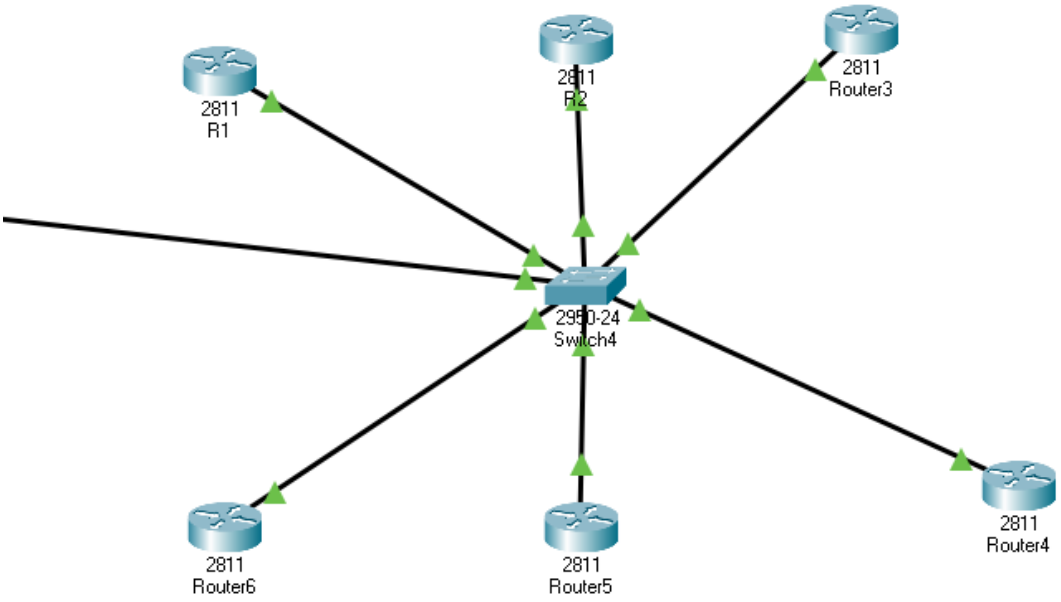
Network Infrastructure:

"The multiaccess network at the ISP Head Quarter consists of six routers, one switch, and a TACACS+ server. The routers are configured with the OSPF routing protocol, which allows for intelligent routing of traffic and ensures that the network remains available even if one of the routers goes down. The switch is responsible for connecting all of the devices on the network and providing access to the TACACS+ server, which is used to monitor and authenticate network traffic.

Here is the network layout

Network 1	Network 2	Network 3	Network 4	Network 5	Network 6
Subnet: 192.168.0.0/29	Subnet: 192.168.0.8/29	Subnet: 192.168.0.16/29	Subnet: 192.168.0.24/29	Subnet 192.168.0.32/29	Subnet: 192.168.0.40/29
Router 1: 192.168.0.1	Router 2: 192.168.0.9	Router 3: 192.168.0.17	Router 4: 192.168.0.25	Router 5: 192.168.0.33	Router 6: 192.168.0.41
Host 1: 192.168.0.3	Host 3: 192.168.0.10	Host 5: 192.168.0.18	Host 7: 192.168.0.26	Host 9: 192.168.0.34	Host 11: 192.168.0.42
Host 2: 192.168.0.4	Host 4: 192.168.0.11	Host 6: 192.168.0.17	Host 8: 192.168.0.27	Host 10: 192.168.0.35	Host 12: 192.168.0.43

The routers in



The network has a physical topology of a star, with all devices connected to the central switch. The logical topology is a mesh, with each router able to communicate directly with all other routers on the network. The hardware and software being used on the network includes Cisco routers and switches, running the IOS operating system."

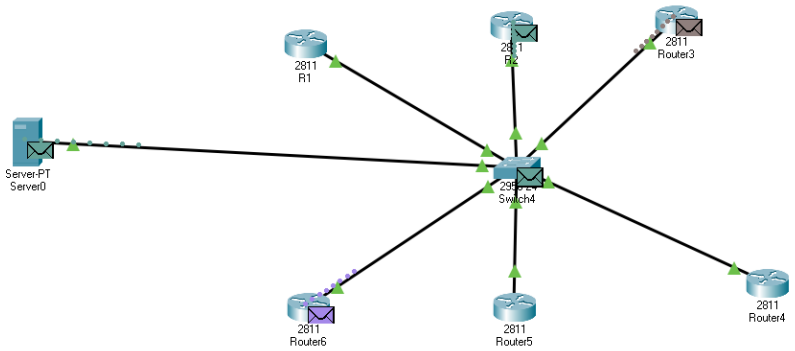
Security Measures:

"To ensure the confidentiality, integrity, and availability of the network, several security measures have been implemented. Switch port security has been configured using sticky MAC addresses, which allows only devices with MAC addresses that have been previously connected to the switch to access the network. This helps to prevent unauthorized access and ensure that only authorized devices can communicate on the network.

```
Switch>ena
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int fa0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security mac-address sticky
Switch(config-if)#
```

In addition, a TACACS+ server has been added to the network to monitor and authenticate network traffic. This allows for centralized control and management of user access and ensures that only

authorized users can access the network resources. Packets are also being encrypted using MD5 to ensure confidentiality and protect against potential threats such as man-in-the-middle attacks.



On R1

R1(config)#ntp server 192.168.1.0

R1(config)#service timestamps log datetime msec

R1(config)#logging host 192.168.1.0

R1(config)#username test secret test1

R1(config)#tacacs-server host 192.168.1.0

R1(config)#tacacs-server key tacacspa55

R1(config)#aaa new-model

R1(config)#aaa authentication login default group tacacs+ local

R1(config)#line console 0

R1(config-line)#login authentication default

On the Tacas server

Client Name: R1

Client IP: 192.168.0.1

Secret: tacaspa1

Server type tacas

Username: test

Password: test1

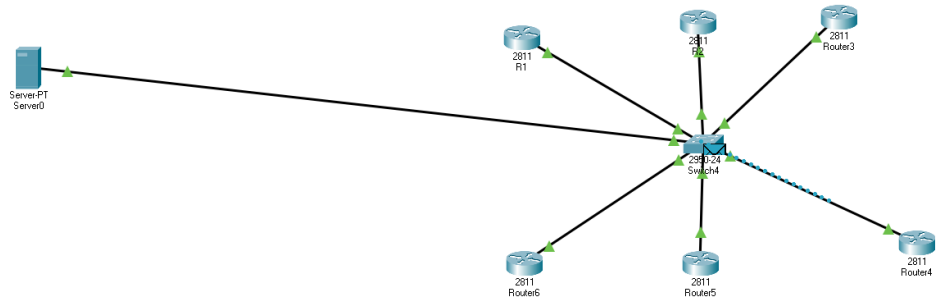
On the switch:

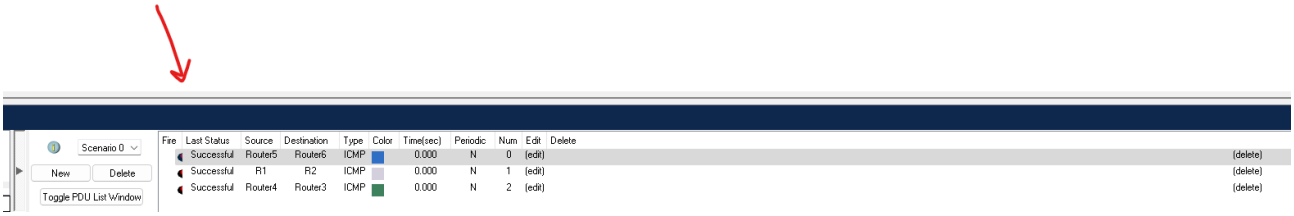
switch(config)#monitor session 1 source interface fastEthernet 0/1

switch(config)#monitor session 1 destination interface fastEthernet 0/7

To maintain the integrity and availability of the network, the OSPF routing protocol has been configured to handle the case where a router goes down. In this scenario, the OSPF protocol will calculate the next designated router and ensure that the network remains available. This helps to ensure that the network is resilient and can continue to support business operations even in the event of a disruption."

Here is a image to confirm that the routers communicate with each other and respond with the respectable packets.





OSPF Configuration:

Router 1:

router ospf 1

router-id 1.1.1.1

network 192.168.2.0/24

network 10.10.11.0/26

network 192.168.11.0/28

network 172.16.2.0/24

Router 2:

router ospf 1

router-id 2.2.2.2

network 192.168.3.0/24

Router 3:

router ospf 1

router-id 3.3.3.3

network 30.30.31.0/24

network 192.168.31.0/25

network 192.168.31.128/25

network 192.168.2.0/24

Router 4:

router ospf 1

router-id 4.4.4.4

network 192.168.4.0/24

network 192.168.41.0/24

network 172.16.41.0/28

Router 5:

router ospf 1

router-id 5.5.5.5

network 192.168.5.0/24

Router 6:

router ospf 1

router-id 6.6.6.6

network 192.168.6.0/24

network 172.16.61.0/24

network 192.168.61.0/25

VLAN 88:

Part 2

I have setup all 6 routers with the these ospf configuration:

Router 1 :

router ospf 1

router-id 1.1.1.1

network 192.168.1.0 0.0.0.255

network 10.10.10.0 0.0.0.63

network 192.168.10.0 0.0.0.15

network 172.16.1.0 0.0.0.255

Router 2:

router ospf 1

router-id 2.2.2.2

network 192.168.1.0 0.0.0.255

Router 3:

router ospf 1

router-id 3.3.3.3

network 30.30.30.0 0.0.0.255

network 192.168.30.0 0.0.0.127

network 192.168.30.0 0.0.0.127

network 192.168.1.0 0.0.0.255

Router 4

router ospf 1

router-id 4.4.4.4

network 192.168.1.0 0.0.0.255

network 192.168.40.0 0.0.0.255

network 172.16.40.0 0.0.0.31

Router 5

router ospf 1

router-id 5.5.5.5

network 192.168.1.0 0.0.0.255

Router 6

router ospf 1

router-id 6.6.6.6

network 192.168.1.0 0.0.0.255

network 172.16.60.0 0.0.0.255

network 192.168.60.0 0.0.0.127

Here are 5 positive things about using OSPF (Open Shortest Path First):

It is a link-state routing protocol, meaning it has a complete understanding of the entire network topology and can therefore make more intelligent routing decisions.

OSPF is highly scalable and can be used in large enterprise networks.

It supports multi-area networks, allowing for better organization and management of large networks.

OSPF has a low overhead and is efficient in terms of bandwidth usage.

It supports equal-cost multi-path routing, allowing for increased redundancy and reliability.

Here are 5 negative things about using OSPF:

It can be more complex to configure and manage than other routing protocols, such as EIGRP.

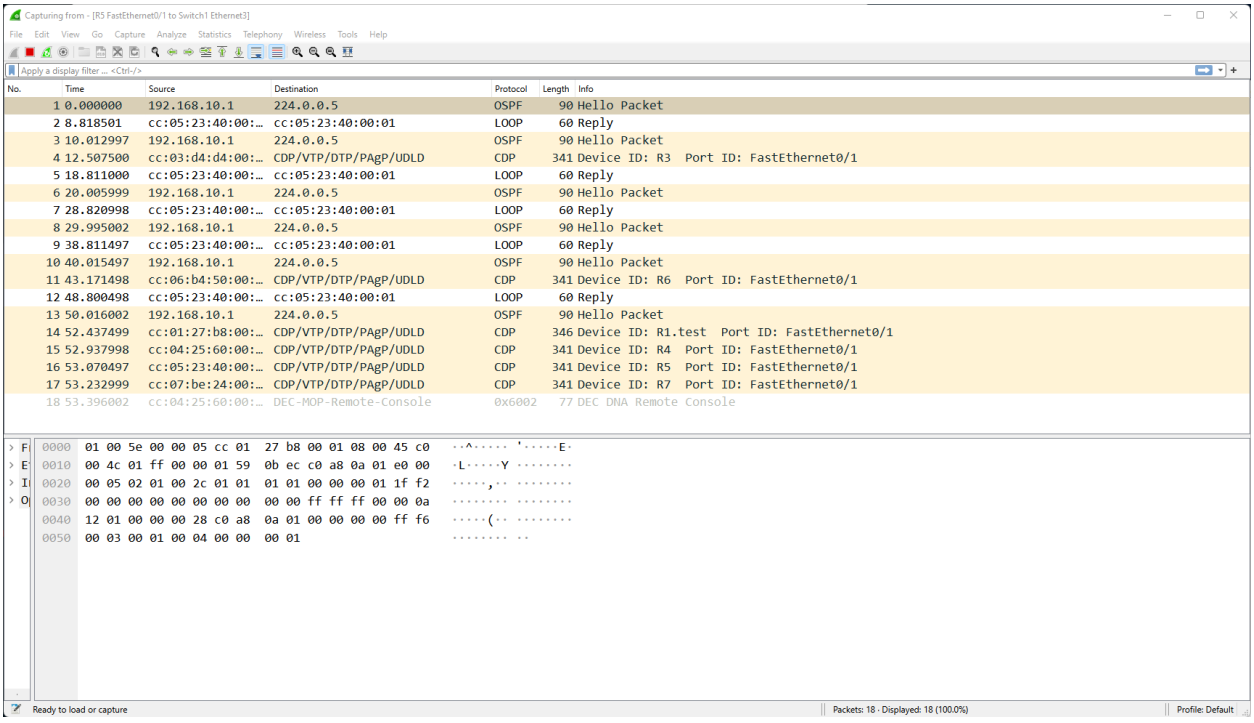
OSPF requires more CPU and memory resources than other routing protocols.

It can take longer for OSPF to converge after a topology change compared to other protocols.

OSPF does not support authentication by default, making it vulnerable to routing attacks.

OSPF does not support route tagging, making it difficult to manipulate routing decisions based on specific criteria.

Here is a screenshot of the ospf configuration and how the routers communicate with eachother:



Here is a description about my code:

The change_config function takes in a crt object and a VLAN number as arguments and sends a command to change the VLAN configuration to the specified VLAN.

The change_dr function takes in a crt object as an argument and sends a command to change the Designated Router (DR) on the shared network.

The detect_threat function takes in a crt object as an argument and sends several commands to display the MAC addresses, OSPF routing information, and network control PDUs on the shared network. It then

checks for unknown MAC addresses, other than OSPF routing information, or other than network control PDUs and returns True if any of these are present.

The main function establishes a connection to routers and switches using the pycrt module and the CRT class, and then calls the change_dr and detect_threat functions. If the detect_threat function returns True, indicating that a threat has been detected, the main function calls the change_config function to change the operating VLAN and forward a copy to VLAN 88. Finally, the function disconnects from the routers and switches.

```
import pycrt  # import the pycrt module


# Print message to indicate the start of the program
print ("PROGRAM STARTED")


def change_vlan_config(crt, vlan):
    """
    Changes the VLAN configuration of the switch to the specified VLAN.
    Parameters:
        crt (obj): an object representing a connection to a router or switch
        vlan (str): the VLAN number to be configured
    """
    # Send command to change the VLAN configuration
    crt.Send(f"configure vlan {vlan}\n")


def change_dr(crt):
    """
    Changes the Designated Router (DR) on the shared network.
    Parameters:
        crt (obj): an object representing a connection to a router or switch
```

```
    """
    # Send command to change the DR
    crt.Send("change dr\n")

def detect_threat(crt):
    """
    Detects a threat on the shared network by looking
    # Send command to display MAC addresses on the shared network
    crt.Send("display mac-address\n")
    mac_addresses = crt.ReadString()
    print("Threat detected on the shared network!!!!")

    # Check if any unknown MAC addresses are present
    if "unknown" in mac_addresses:
        return True

    # Send command to display OSPF routing information
    crt.Send("display ospf routing-table\n")
    ospf_routing = crt.ReadString()

    # Check if any other than OSPF routing information is present
    if "unknown" in ospf_routing:
        print("Another protocol has been detected.")
        return True

    # Send command to display network control PDUs
    crt.Send("display network-control-pdu\n")
    pdus = crt.ReadString()
```

```
    # Check if any other than network control PDUs are present
    if "unknown" in pdus:
        return True

    return False

def main():
    # Connect to the routers and switches using SecureCRT
    crt = pycrt.CRT()
    crt.Connect("192.168.10.1", "ipcisco", "ipcisco_1 ")
    crt.Connect("192.168.10.2", "ipcisco", "ipcisco_1 ")

    # Change the DR on a daily basis
    change_dr(crt)

    # Check for threats on the shared network
    if detect_threat(crt):
        # Change the operating VLAN and forward a copy to VLAN 88
        print("A threat has been sent to vlan 88")
        change_vlan_config(crt, "88")

    # Disconnect from the routers and switches
    crt.Disconnect()

# Call the main function to start the program
main()
```

To be able to connect to each router we have to enable ssh on each router. Here is an example on how such a setup would look like:

Router# conf t - This command enters global configuration mode on the router.

R1(config)# username test password test1 - This command creates a new user named "test" with the password "test1".

R1(config)# line vty 0 7 - This command enters line configuration mode for virtual terminal lines 0 through 7.

R1(config-line)# transport input ssh - This command enables SSH as an allowed protocol for incoming connections on the virtual terminal lines.

R1(config-line)# access-class 1 in - This command applies access list number 1 to incoming connections on the virtual terminal lines.

R1(config-line)# login local - This command enables local login authentication on the virtual terminal lines.

R1(config)# access-list 1 permit 192.168.1.0 0.0.0.255 - This command creates an access list that permits incoming connections from the IP address range 192.168.1.0 to 192.168.1.255.

R1(config)# interface fa0/0 - This command enters interface configuration mode for interface GigabitEthernet0/0.

R1(config-if)# ip address 192.168.1.1 0.0.0.255 - This command assigns the IP address 192.168

R1(config-if)# no shutdown

R1(config)# enable password cisco

R1(config)# ip domain-name test

R1(config)# crypto key generate rsa

Then you can choose any number that is bigger than 512 if you chose a number bigger than 800 you get ssh 1.99

Secure Shell (SSH) is a network protocol that allows secure remote login to a computer over an unsecured network. It is commonly used to securely access servers, network devices, and other computers over the internet.

When you establish an SSH connection to a remote host, you are able to enter commands and receive responses just as if you were sitting at the physical computer. This is useful for tasks such as remotely administering servers, managing network devices, and transferring files between computers.

One of the main benefits of SSH is that it uses strong encryption to secure the connection and protect the data being transmitted. This makes it more secure than other methods of remote access, such as Telnet, which does not encrypt data and is therefore vulnerable to interception.

SSH connections are established using a client-server model. The client is the computer you are using to connect to the remote host, and the server is the computer you are connecting to. To establish an SSH connection, you need to use an SSH client program, such as the built-in Terminal program on macOS or Linux, or a third-party program like PuTTY on Windows.

On the server side, an SSH server program must be running and listening for incoming connections. When you initiate an SSH connection, the client and server exchange keys to establish a secure connection, and then you are prompted to enter your login credentials (usually a username and password). Once you have authenticated, you can enter commands and receive responses from the server just as if you were sitting at the physical computer.

In summary, the ISP Head Quarter Manager has consulted me, a professional network administrator, to secure the multiaccess network at the ISP Head Quarter. To achieve this, I have implemented a range of security measures including the use of the OSPF routing protocol, switch port security, TACACS+ server, and packet encryption. The network infrastructure consists of six routers, one switch, and a TACACS+ server, all connected in a star physical topology and a mesh logical topology. The network uses Cisco routers and switches running the IOS operating system. To ensure the confidentiality, integrity, and availability of the network, I have implemented switch port security using sticky MAC addresses and a TACACS+ server for centralized control and management of user access. Packets are also encrypted using MD5 to protect against potential threats. I have also configured logging on the routers to track events and monitor the network, as well as implemented access control lists to allow or deny traffic based on specified criteria. Overall, these security measures help to protect the multiaccess network at the ISP Head Quarter and ensure that it remains secure and available to authorized users.