NCCD-3: Network Traffic Analysis

April 2024

u2261809

Contents:

1. Network Discovery:

1.1 Time & Period of Data Collection:

The time of a specific packet capture can be gleamed from the .pcap file. This is because the time of packet reception is logged by the capturing NIC (Network Interface Card) or the packet capturer's driver. To find this information we can look at the *Time* column in WireShark. This column shows the amount of seconds lapsed since the start of the capture. By sorting this column it is trivial to identify the first packet sent, as it alone has a time of *0.000000*. The packet summary is as follows:

```
1      0.000000    10.10.10.20 10.10.10.10 S7COMM      153   ROSCTR:[Job     ] Function:[Read Var]
```

Figure 1: Packet Number 1

Looking inside this Figure 1 packet we can see the arrival time:

```
Oct 21, 2015 23:10:34.995270000 GMT Daylight Time
```

We can then filter in reverse, sorting by the latest packet received, the summary of this packet and its arrival time is:

```
2274747     54422.093993      77.245.33.76      10.100.152.128    ESP   126   ESP (SPI=0xa30ff23a)
```

Figure 2: Packet Number2274747

```
Oct 22, 2015 14:17:37.089263000 GMT Daylight Time
```

Using this information the following can be calculated, shown in Table 1:

Start of capture	Oct 21, 2015 23:10:34
End of capture	Oct 22, 2015 14:17:37
Duration of capture	54422.093993 seconds = 15 hours, 7 minutes, and 2 seconds

Table 1: Finalised Transmission Timing Information

Table Key:
Private IP’s
Public IP’s

1.2 IPv4 Address Enumeration

To identify all the IPv4 addresses contained within the capture, we can filter in WireShark, select Statistics > IPv4 Statistics > All Addresses. The device manufacturer can also be found, in the both packet info and through MAC addresses, these values, alongside relevant device info have been tabulated here:

IPv4 Address	Additional Information
192.168.2.64	Westermo Network Technologies
192.168.2.53	Westermo Network Technologies - Android
192.168.2.44	Westermo Network Technologies
192.168.2.22	Westermo Network Technologies
192.168.2.21	Westermo Network Technologies
192.168.2.199	Westermo Network Technologies
192.168.2.166	Westermo Network Technologies - Windows
192.168.2.137	Westermo Network Technologies - Apple IOS
192.168.2.133	Westermo Network Technologies
192.168.2.110	Westermo Network Technologies
192.168.143.254	Westermo Network Technologies
192.168.143.155	Apple Inc
192.168.143.1	Virtual Machine - VMware
192.168.1.79	Siemens - Rugged Comm Inc
192.168.1.71	Siemens - Rugged Comm Inc
192.168.1.68	Westermo Network Technologies
192.168.1.2	Westermo Network Technologies
192.168.1.10	Ubiquity Inc- Windows
192.168.0.3	Westermo Network Technologies
173.252.90.4	Unknown Vendor - HTTPS Web Server
172.16.184.40	Ubiquity - Remote Administration
17.253.54.251	Westermo Network Technologies
17.253.34.253	Siemens - Rugged Comm Inc
17.130.137.75	Unknown Vendor
17.130.137.73	Unknown Vendor
17.110.230.30	Unknown Vendor
17.110.224.213	Unknown Vendor
141.82.217.52	Unknown Vendor
108.160.163.110	Unknown Vendor
10.218.104.244	Siemens - Rugged Comm Inc
10.100.159.27	Apple Inc
10.100.159.253	Samsung Electro-Mechanics
10.100.159.247	Apple Inc
10.100.159.228	Apple Inc
10.100.159.227	Hon Hai Precision
10.100.159.218	Sony Corporation
10.100.159.207	Samsung Electro-Mechanics
10.100.159.151	Apple Inc
10.100.159.125	OnePlus Tech
10.100.158.185	Apple Inc
10.100.158.168	Microsoft Corporation
10.100.152.15	Apple Inc
10.100.152.128	Innominate Security Technologies
10.100.152.119	Apple Inc
10.100.152.10	Cisco Systems
10.10.10.30	Wistron InfoComm - Windows
10.10.10.20	Siemens Numerical Control
10.10.10.10	Siemens AG - Siemens
192.168.88.2	Siemens - Rugged Comm Inc
192.168.88.15	HOST ENGINEERING
192.168.88.130	MOXA Technologies - ICS_Device
192.168.88.115	DigiBoard - Linux
192.168.88.105	CIMSYS Inc
192.168.88.100	HOST ENGINEERING
192.168.88.1	Westermo Network Technologies
192.168.57.3	COMPAL Information
192.168.57.2	Unknown
93.158.94.210	Westermo Network Technologies
93.158.110.218	Unknown
93.158.110.200	Unknown
83.140.27.11	Unknown
8.8.8.8	Siemens - Rugged Comm Inc
77.245.33.76	Unknown - Hostname: machine-gw1.stage1.mguard.com
74.125.205.188	Unknown
54.241.179.26	Unknown
54.210.217.83	Unknown
52.5.95.205	Unknown
52.4.151.114	Unknown
21.2.2.2	Westermo Network Technologies
199.16.156.72	Unknown
199.16.156.70	Unknown
199.16.156.48	Unknown
199.16.156.231	Unknown
199.16.156.198	Unknown
193.209.237.4	Unknown
193.182.190.178	Unknown
192.195.142.14	Westermo Network Technologies
192.195.142.13	Westermo Network Technologies
192.168.89.2	PEGATRON CORPORATION
192.168.89.1	Siemens - Rugged Comm Inc
192.168.88.95	Siemens - Rugged Comm Inc - Siemens
192.168.88.85	Hi-flying electronics
192.168.88.80	MOXA Technologies
192.168.88.75	Hirschmann Automation and Control - ICS_Device
192.168.88.61	MOXA Technologies - ICS_Device
192.168.88.60	MOXA Technologies - ICS_Device
192.168.88.55	Apple Inc - Kali Linux
192.168.88.54	Apple Inc
192.168.88.53	Apple Inc - Apple IOS
192.168.88.52	Apple Inc
192.168.88.51	ADVANTECH CO - ICS_Device
192.168.88.50	Red Lion Controls - ICS_Device
192.168.88.49	AXIS Communications - Linux
192.168.88.30	Siemens Numerical Control - Siemens
192.168.88.254	Innominate Security Technologies
192.168.88.25	ADVANTECH CO - ICS_Device
192.168.88.20	PHOENIX CONTACT Electronics - ICS_Device

Table 2: Full IPv4 Address Listing

1.3 Device Identification

All devices have now been tabulated, now to begin identification.

First devices with identical MAC addresses can be combined, as MAC addresses are unique to a given device, due to it partially containing the device's serial number.
Identification is provided through the combination of both the device manufacturer (MAC address) and the packets sent to the given IP as seen in WireShark. We will now step through each device, explaining the functionality of each IP assigned.

MAC Address	**000ADC6485C2**
Implied Manufacturer	RuggedCom Inc (Siemens)
IP Address	Implied Functionality
8.8.8.8	Google public DNS server (NAT Copying)
10.128.104.244	ISATAP IPv4 → IPv6 protocol bridge
17.253.34.253	Public NTP Server
192.168.1.71	Open port 5000 typically Upnp
192.168.1.79	Open port 5000 typically Upnp
192.168.89.1	Router

Table 3: 000ADC6485C2 MAC Analysis

Address: 8.8.8.8

```
192.168.89.2    8.8.8.8    DNS    74    Standard query 0x0002 A ntp1.dlink.com
```

Figure 3: Packet Number 8

Here, in Figure 3, we can see a private IP performing a DNS request. This misleadingly appears to share a MAC address with our router. However this is potentially due to NAT copying the MAC address of the final private device touched on both reception and transmission of traffic, being our router.

Address: 10.218.104.244

```
192.168.88.51    10.218.104.244    NBNS    92    Name query NB ISATAP<00>
```

Figure 4: Packet Number 1983911

This device, as seen in Figure4, is handling NBNS name resolution, information relating to potential IPv4 to IPv6 protocol bridging using ISATAP.

1.3 Device Identification

Address: 17.253.34.253

```
192.168.89.2      17.253.34.253      NTP    90    NTP Version 4, client
```

Figure 5: Packet Number 1783481

This IP, as seen in Figure 5, is clearly functioning as an NTP sever, a crucial service for time synchronicity between network devices.

Address: 192.168.1.71

```
192.168.89.2      192.168.1.71      TCP    78    [TCP Retransmission] 53526 → 5000 [SYN] Seq=0
Win=65535 Len=0 MSS=1460 WS=32 TSval=1131374391 TSecr=0 SACK_PERM
```

Figure 6: Packet Number 1787249

In Figure 6, a 3 way TCP handshake can be seen, with 192.168.89.2 attempting a connection to a network service hosted on port 5000 on the 192.168.1.71 machine. This is most likely TCP Universal Plug and Play (UPnP). This service, hosted on port 5000 by default.

Address: 192.168.1.79

```
192.168.89.2      192.168.1.79      TCP    78    [TCP Retransmission] 53525 → 5000 [SYN] Seq=0
Win=65535 Len=0 MSS=1460 WS=32 TSval=1131367175 TSecr=0 SACK_PERM
```

Figure 7: Packet Number 1786367

As Figure 7 shows, much like the previous address (192.168.1.71) there is an open port 5000.

Address: 192.168.89.1

```
192.168.89.1      192.168.89.2      ICMP   102   Destination unreachable (Network unreachable)
```

Figure 8: Packet Number 9

Figure 8 shows that the address 192.168.89.1 is transmitting destination unreachable alert packets to network IPs. This implies that the 192.168.89.1 interface is functioning as a router or gateway, involved in network management handling and reporting on network trafficking issues.

1.3 Device Identification

Moving on to the next device:

MAC Address	**28CFE818B5ED**
Implied Manufacturer	Unkown
IP Address	Implied Functionality
10.0.1.82	Apple Device, probable endpoint
10.100.152.92	Apple Device, probable endpoint

Table 4: Apple MAC breakdown

Despite not sending traffic, the NIC provider, Apple Inc, hints to the functionality of these devices as user endpoints.

MAC Address	**0418D683DB16**
Implied Manufacturer	Ubiquity Inc
IP Address	Implied Functionality
10.10.10.1	Siemens local device monitoring
172.16.184.40	Admin system used for remote access
192.168.1.10	Industrial (Siemens) automation controller

Table 5: Ubiquity MAC breakdown

This device, produced by Ubiquity, is configured to manage and interact with Siemens equipment, this aligns with the presence of several Siemens devices on the network, alongside numerous other industrial control devices.

Address: 10.10.10.1

This address exclusively queries 10.10.10.10. This machine is a Siemens device. This indicates a function as a Siemens administrative and monitoring device.

1.3 Device Identification

Continuing with the Ubiquity device:

Address: 172.16.184.40

```
10.10.10.30 172.16.184.40     VNC   1027
```

Figure 9: Packet Number 785923

This address is clearly used for remote control, as seen by the usage of Virtual Network Computing (VNC) service, hosted on port 5900 as seen in the packet information section of Figure 9. It is being controlled by what may be an administrative endpoint hosted at 10.10.10.30.

Address: 192.168.1.10

```
192.168.1.10      10.10.10.10 S7COMM      192   ROSCTR:[Job     ] Function:[Setup
communication] | ROSCTR:[Job     ] Function:[Read Var] | ROSCTR:[Job     ] Function:
[Write Var]

   10.10.10.10 192.168.1.10      COTP  76    CC TPDU src-ref: 0x0007 dst-ref: 0x0001
```

Figure 10: Packet Number 1260388 And 1260389

Functionally this address is active in industrial automation controlling, evident through its usage of protocols such as COTP and S7Comm as seen in Figure 10. This Siemens proprietary protocol (S7Comm) is designed to interact with Siemens Programmable Logic Controllers (PLC), with COTP used in information transportation.

1.3 Device Identification

MAC Address	**A2F4010001D6**
Implied Manufacturer	Intentionally Obscured
IP Address	Implied Functionality
10.100.152.1	Router & Firewall administrator
74.125.205.188	Google public services (NAT)
52.4.151.114, 52.5.95.205, 54.210.217.83, 54.241.179.26	Amazon Web Services (AWS) (NAT)
17.110.224.213, 17.110.230.30, 17.130.137.73, 17.130.137.75	Apple Inc, potential cloud services (NAT)
93.158.110.218, 93.158.94.210	Yandex Russian search & services (NAT)
77.245.33.76, 83.140.27.11	ILSS Logistics German cloud services 83.140.27.11 - DNS server (NAT)
199.16.156.48, 199.16.156.70, 199.16.156.72, 199.16.156.198, 199.16.156.231	Twitter social media and API services (NAT)
193.182.190.178, 193.209.237.4	Nixu Finish IT security provider (NAT)
173.252.90.4	Facebook social media (NAT)
108.160.163.110	Dropbox cloud storage provider (NAT)

Table 6: Obscured MAC breakdown

Based on this list of IPs, full breakdowns for each one is redundant, as it is apparent most of these IPs don’t provide a service local to our network. They appear here much like 8.8.8.8 did on our 192.168.89.1 router, this is a product of NAT copying the MAC address of our router onto all inbound and outbound traffic.

Additional evidence for this is the lack of data sent to the router itself, if the router was not using NAT, we would expect to see abundant inbound and outbound traffic.

1.3 Device Identification

Address: 74.125.205.188

```
74.125.205.188    10.100.159.227    TLSv1.2    85    Application Data
```

Figure 11: Packet Number 2268509

```
74.125.205.188    10.100.159.227    TCP    60    5228 → 55496 [FIN, ACK] Seq=32 Ack=1 Win=361 Len=0
```

Figure 12: Packet Number 2270740

As seen in Figure 11 and 12 we can see a connection between a device hosted at 10.100.159.227 and a Google pubic service connected by 74.125.205.188. This connection has not been blocked, and has been routed successfully, but as we will see soon, this is not always the case.

Address: 173.252.90.4

```
173.252.90.4    10.100.159.253    TLSv1.2    97    Encrypted Alert
```

Figure 13: Packet Number 2234500

```
173.252.90.4    10.100.159.253    TCP    66    443 → 56685 [RST, ACK] Seq=32 Ack=1 Win=65 Len=0
TSval=3023931834 TSecr=3993093
```

Figure 14: Packet Number 2234501

Figure 13 and 14 routed to Facebook located at 173.252.90.4 is dropped and terminated early, seen in Figure 14. This shows a potential function as a firewall.

MAC Address	**00077C1A6183**
Implied Manufacturer	Westermo Technologies
IP Address	Implied Functionality
21.2.2.2	USA DoD Network information centre
192.168.88.1	Addressable Router Address
93.158.94.210	Epm Data Swedish IT & Cloud operations
192.168.0.x	1x Independent subnet (NAT Copying)
192.168.1.x	2x Independent subnet (NAT Copying)
192.168.2.x	10x Independent subnet (NAT Copying)
192.168.143.x	1x Independent subnet (NAT Copying)

Table 7: Westermo MAC breakdown

An entire 192.168.2.x subnet appears here rather than independently implies NAT, hinting at a router. Additionally a private IP address 21.2.2.2 alludes to an internet connection. The logical router address would be 192.168.88.1

Address: 192.168.2.199

```
192.168.2.199    192.168.88.75    TCP    62    53005 → 56210 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
```

Figure 15: Packet Number 1108187

Figure 15 shows a TCP packet being forwarded aligning with the traditional functionality of a router.

1.3 Device Identification

Address: 192.168.2.199

```
192.168.88.60     192.168.2.133     TCP   66    502 → 915 [ACK] Seq=1 Ack=46 Win=17376
Len=0 TSval=235014849 TSecr=146629
```

Figure 16: Packet Number 464082

Figure 16 again shows routing, again, however this occurs from a different address in the device to a switch (justified later) at 192.168.88.60.

Device Type	Summary	Reasoning
Router	192.168.88.1	This address is typical of a router, follows the network's naming convention as seen earlier
Router	DNS Functionality	192.168.88.1 hosts a lot of DNS traffic, this function is indicative of a higher layer switch or router
Router	Routing Functionality	This device has been seen in both figure 15 & 16 as functioning similar to a router, by performing forwarding decisions based upon IP addresses.

Table 8: Router Justification

MAC Address	**A0999B1CD865**
Implied Manufacturer	Apple Inc
IP Address	Implied Functionality
192.168.88.53	Virtual iOS endpoint
192.168.88.54	Apple user endpoint
192.168.88.55	Virtual Kali Linux virtual machine
192.168.143.155	Virtual user endpoint

Table 9: Apple MAC breakdown, again

Table 9 appears to show an endpoint device, several OS's implies virtualisation. This conclusions also explains the address 192.168.143.155 which appears to span a subnet, something unusual for an endpoint, however upon inspecting traffic sent from this address, it exclusively contacts a DNS server at 192.1688.143.1. Looking into this address further we can see that it is in-fact virtual, with its NIC provider listed as VM-Ware, a popular virtualisation host.

1.3 Device Identification

This concludes all devices with more than one visible IP. All other devices have a single IP, tabulated here. The manufacturer comes from the MAC address:

10.x.x.x Addresses:

Address	Manufacturer	Function	Justification
10.0.1.82	Apple	User Endpoint	Apple produce end-user devices
10.10.10.10	Siemens	Programmable Logic Controller	Receives and sends instructions of S7Comm a protocol designed for Siemens' PLCs
10.10.10.20	Siemens	CNC Machine	This machine was produced by Siemens but more specifically their numerical control division, producing CNC manufacturing devices
10.10.10.30	Wistron InfoComms	User Endpoint	Wistron produces a variety of devices however this device is running Windows, indicating its one of their "All-in-one" endpoint systems
10.100.152.10	Cisco	DHCP Server	Seen sending DHCP packets around the network
10.100.152.15	Apple	User Endpoint	Apple produce end-user devices
10.100.152.119	Apple	User Endpoint	Apple produce end-user devices
10.100.152.128	Innomate Security Technologies	VPN Server	This is justified in more detail further on in the report.
10.100.158.168	Microsoft	User Endpoint	Microsoft produce many devices typically in the consumer sphere
10.100.158.185	Apple	User Endpoint	Apple produce end-user devices
10.100.159.27	Apple	User Endpoint	Apple produce end-user devices
10.100.159.125	OnePlus	User Endpoint	OnePlus produce user phones
10.100.159.151	Apple	User Endpoint	Apple produce end-user devices
10.100.159.207	Samsung Electro-Mechanics	Embedded Device	A subsidiary of Samsung, producing embedded computing devices
10.100.159.218	Sony Corporation	User Endpoint	A producer of phones and tablets
10.100.159.227	HonHai Corporation	User Endpoint	HonHai produces products for Apple and Apple produce end-user devices
10.100.159.228	Apple	User Endpoint	Apple produce end-user devices
10.100.159.247	Apple	User Endpoint	Apple produce end-user devices
10.100.159.253	Samsung Electro-Mechanics	Embedded Device	A subsidiary of Samsung, producing embedded computing devices

Table 10: 10.X.X.X Addresses

1.3 Device Identification

Address: 10.100.152.10

```
10.100.152.10     10.100.152.128    DHCP  345   DHCP Offer    - Transaction ID 0x7e82fa7d
```

Figure 17: Packet Number 2239447

This packet shows evidence of 10.100.152.10 acting as a DHCP server. Critical for dynamically assigning IP addresses to new device connections. However not many DHCP packets sent. DHCP is therefore done on the switches.

Address: 10.100.152.128

```
10.100.152.128    77.245.33.76      ISAKMP      570   Identity Protection (Main Mode)
```

Figure 18: Packet Number 2239461

This packet shows evidence of 10.100.152.128 acting as a VPN server, more specifically as an Internet Key Exchange (IKE) server. This is part of the IPsec protocol suite, frequently mentioned in this device’s packets.

1.3 Device Identification

192.168.x.x Addresses:

Address	Manufacturer	Function	Justification
192.168.0.2	Apple	User Endpoint	Apple produce end-user devices
192.168.57.2	Unknown	Secure SSH server	This device exclusively sends SSH traffic
192.168.57.3	Compal Corporation	User Endpoint	This device is exclusively an SSH client and is therefore a user endpoint
192.168.88.2	RuggedCom Inc	ICS Device	This system is running an HTML management service, exact device type is not clear
192.168.88.15	Host Engineering	Programmable Logic Controller	This device hosts management services on HTML, Host Engineering produces automatising PLC units
192.168.88.20	Phoenix Contact	Bus Coupler	OS fingerprinted as Phoenix TCP Bus Coupler
192.168.88.25	Advantech	Programmable Logic Controller	OS fingerprinted as a Advantech PLC, Advantech make automation technologies
192.168.88.30	Siemens	Programmable Logic Controller	OS fingerprinted as a Siemens PLC, running Siemens OS adds credibility
192.168.88.49	Axis Comms	CCTV Cameras	Producer of network CCTV cameras
192.168.88.50	Red Lion Controls	DSP Converter	OS fingerprinted as Red Lion DSP
192.168.88.51	Johnson Controls	Network Controller	OS fingerprinted as MS-NAE3510-2 a Metasys MS series network controller
192.168.88.52	Apple	User Endpoint	Apple produce end-user devices
192.168.88.60	MOXA	Ethernet Switch	Fingerprinted as an EDS switch, visible files in capture confirming model as an EDS-516A/EDS-508A switch.
192.168.88.61	MOXA	Ethernet Switch	Fingerprinted as an EDS switch, visible files in capture confirming model as an EDS-516A/EDS-508A switch.
192.168.88.75	Hirschman Automation	Industrial Firewall or VPN-Router	OS fingerprinted as a EAGLE 20 TOFINO, a discontinued industrial firewall.
192.168.88.80	MOXA	Ethernet Switch	Fingerprinted as an EDS switch, visible files in capture confirming model as an EDS-516A/EDS-508A switch.

Table 11: 192.168.X.X addresses [1]

1.3 Device Identification

192.168.x.x Addresses:

192.168.88.85	Hi-Flying Electronics	IoT Device	Hi-Flying Electronics produces industrial IoT devices
192.168.88.95	RuggedCom Inc	Serial Device Server & Managed Ethernet Switch	The OS has been fingerprinted as a RuggedCom RS910 series, this product acts as both a managed switch as well as a hardened serial device server
192.168.88.100	Host Engineering	Unidentified	Too little information to conclusively identify the device type, likely a PLC but is not conclusive
192.168.88.105	CIM Sys	ISP Device	No conclusive information, after doing online research, this device is provided by the network's ISP for remote modem administration.
192.168.88.115	Digiboard	Device and Network Monitor	This device produced by Digi, is running a software called Digi Connectware, this installs Linux OS, and allows for monitoring of both network (packets and ports) and hardware conditions.
192.168.88.130	MOXA	Serial Device Server	This device has been fingerprinted as the MOXA NPort 5610. This device allows for the communication of serial devices to a TCP/IP network allowing for managing and accessing potentially older, otherwise obsolete devices over the network.
192.168.88.254	Innonimate Security	Embedded Security Device	Although unclear, devices typically manufactured by this company indicate it is an embedded security device. It also seemingly ran an Nmap scan, however I may be mistaken.
192.168.89.2	Pegatron Corporation	User Endpoint	Pegatron currently makes the iPhone 13 & 14
192.168.143.1	VM Ware	Virtual Router	VM Ware don't produce hardware, therefore this device is virtual

Table 12: 192.168.X.X addresses [2]

1.4 Security Concerns

Address: 10.100.152.1

This IP corresponds to one of the networks main routers. This device has OpenSSH v6.6.1p1 enabled on port 22. This can be proved by an SSH connection between 192.168.88.75 (Industrial Firewall) and router.
This version is outdated and vulnerable to numerous exploits.

Address: 192.168.88.115

This IP corresponds to our DigiBoard device, again with SSH open on port 22. This version is even older being OpenSSH v4.0. This version is incredibly vulnerable, having been released in 2005, is vulnerable to a huge amount of exploits, even at the time of packet capture this system is very outdated.

```
192.168.88.115    192.168.2.137    SSH   86    Server: Protocol (SSH-2.0-OpenSSH_4.0)
```

Figure 19: Packet Number 2240443

1.5 Connection Overview and VLANS

When tasked with reconstructing a network from a .pcap file, it is important to understand that this format primarily displays logical connections, not direct VLAN configurations or physical cabled links. For example VLAN tags and physical pathways, especially those paths mediated by switches, such as on this network, are not visible within this capture.

To rebuild this .pcap, we will have to focus on analysing the IP address allocations, and infer potential VLANs based off typical network structures and segmentations. To create a full accurate reconstruction, additional resources such as network documentation or access to physical network infrastructure will be required.

Subnet: 192.168.88.0/24

Justification for the router is not required as it is contained within this subnet. As for switches, by filtering on WireShark using:

```
ip.dst==192.168.88.0/24 && ip.src==192.168.88.0/24
```

It is see that 192.168.88.61 seems to be the sole switch communicating with the router. This implies that the other switches on the subnet (.61, .80, .95) connect to this switch, in a hierarchical setup. The VLANS, using this information is tabulated here, using VLSM to split the subnet between the three switches :

VLAN 88	
Subnet	192.168.88.0/26
Assigned Switch	192.168.88.60
Router	192.168.88.1
Usable Range	192.168.88.1 to 192.168.88.62
IP Range	.2, .15, .20, .25, .30, .49, .50, .51, .52, .53, .54, .55, .60

VLAN 881	
Subnet	192.168.88.64/27
Assigned Switch	192.168.88.80
Router	192.168.88.1
Usable Range	192.168.88.65 to 192.168.88.94
IP Range	.75, .80, .85

VLAN 882	
Subnet	192.168.88.96/27
Assigned Switch	192.168.88.95
Router	192.168.88.1
Usable Range	192.168.88.97 to 192.168.88.118
IP Range	.100, .105, .115

1.5 Connection Overview and VLANS

Subnet: 192.168.89.0/24

VLAN 89	Grouping by subnet and +1
Subnet	192.168.89.0/24
Assigned Switch	Direct Router Connection
Router	192.168.89.1
IP Range	.1, .2
Broadcast Address	192.168.89.255

Justification not required as router is contained on subnet.

Subnet: 192.168.57.0/24

VLAN 57	Grouping by subnet and +1
Subnet	192.168.57.0/24
Assigned Switch	Assumed Switch: 192.168.57.4
Router	Assumed Router: 192.168.89.1
IP Range	.2, .3
Broadcast Address	192.168.57.255

This device does not appear to communicate outside of its subnet, and shows no evidence of internet connection. This subnet only shows SSH traffic.

Subnet: 192.168.2.0/24

VLAN 2	Grouping by subnet
Subnet	192.168.2.0/24
Assigned Switch	Assumed Switch: 192.168.2.2
Router	192.168.88.1
IP Range	.21, .22, .44, .53, .64, .110, .133, .137, .166, .199
Broadcast Address	192.168.2.255

This subnet contacts the 192.168.88.x subnet through the router 192.168.88.1 and is therefore connected evidence included below in Figure 20.

```
192.168.88.20    192.168.2.137    TCP    60    80 → 41644 [SYN, ACK] Seq=0 Ack=1 Win=4096 Len=0
```

Fig 20: Paket Number 252305

1.5 Connection Overview and VLANS

Subnet: 10.0.1.0/24

VLAN 10	Grouping by subnet
Subnet	10.0.1.0/24
Assigned Switch	Assumed Switch: 10.0.1.2
Router	Assumed Router: 10.100.152.1 But most likely: 10.0.1.1
IP Range	.82
Broadcast Address	10.0.1.255

This device is an endpoint, and does not send any traffic. This device only queries a device at 10.0.1.1, this device based on naming convention would be a router. However this device does not appear to exist, it may be offline at time of capture. Therefore this subnet is not currently possible to model accurately.

Subnet: 10.100.152.0/24

VLAN 152	Grouping by subnet
Subnet	10.100.152.0/24
Assigned Switch	Assumed Switch: 10.100.152.2
Router	10.100.152.1
IP Range	.1, .10, .15, .119, .128
Broadcast Address	10.100.152.255

Justification not required as router is contained on subnet.

1.5 Connection Overview and VLANS

Subnet: 10.100.158.0/24

VLAN 158	Grouping by subnet
Subnet	10.100.158.0/24
Assigned Switch	Assumed Switch: 10.100.158.2
Router	10.100.152.1
IP Range	.168, .185
Broadcast Address	10.100.158.255

We can prove this subnet has internet access due to an HTTP request, in Fig 21 this request stems from a public IP at 93.158.110.218, destined for a private IP address 10.100.158.168. This guarantees a connection to our 10.100.152.1 router due to the address 93.158.110.218 sharing a MAC address with this device, as seen in NetworkMiner.

```
93.158.110.218    10.100.158.168    TCP    60    80 → 55147 [FIN, ACK] Seq=1 Ack=1 Win=924 Len=0
```

Fig 21: Paket Number 2261248

Subnet: 10.100.159.0/24

VLAN 159	Grouping by subnet
Subnet	10.100.159.0/24
Assigned Switch	Assumed Switch: 10.100.159.2
Router	10.100.152.1
IP Range	.27, .125, .151, .207, .218, .227, .228, .247, .253
Broadcast Address	10.100.159.255

We know this subnet has internet access due to 10.100.159.218 making a DNS request to a public IP at 83.140.27.11 it therefore needs a router connection. We know that this connection is to our 10.100.152.1 router due to the address 93.158.110.218 sharing a MAC address with this device as seen in NetworkMiner.

```
83.140.27.11      10.100.159.218    DNS    415    Standard query response 0x124b A
clients2.google.com CNAME clients.l.google.com
```

Fig 22: Paket Number 2239340

1.5 Connection Overview and VLANS

Subnet: 192.168.143.0/24

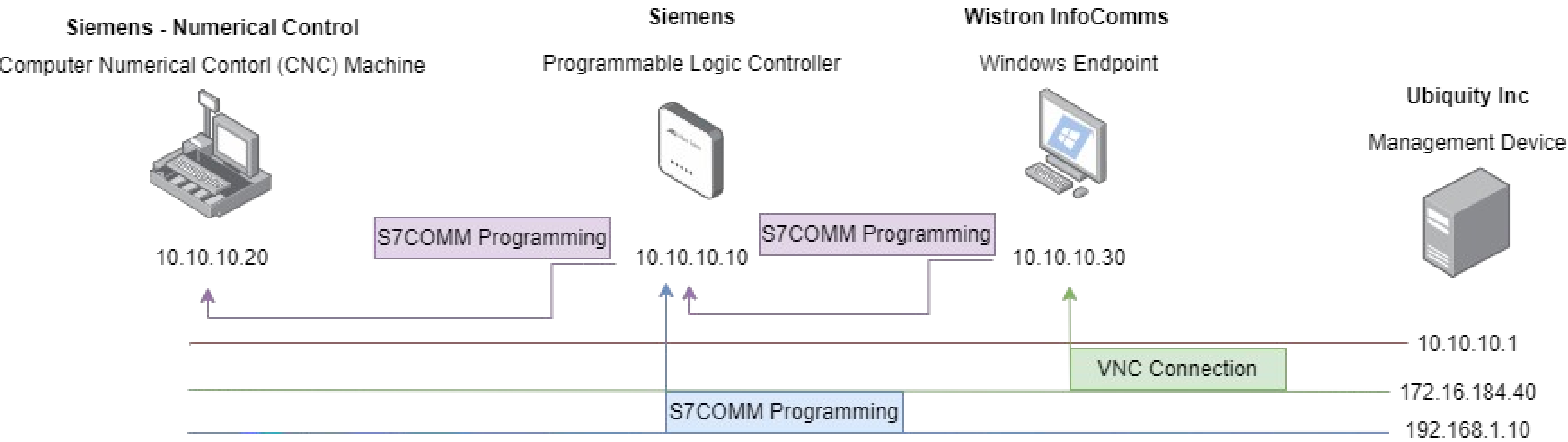
VLAN 143	Grouping by subnet
Subnet	192.168.143.0/24
Assigned Switch	Assumed Switch: 192.168.143.2
Router	Router: 192.168.88.1 also Virtual Router: 192.168.143.1
IP Range	.1, .155, .254
Broadcast Address	192.168.143.255

This subnet has two routers, it shares a MAC address with our 192.168.143.1 but it also seems to contain an address at .1 that has VM Ware as its NIC provider.

Subnet: 10.10.10.0/24

VLAN 100	Grouping by subnet
Subnet	10.10.10.0/24
Assigned Switch	Assumed Switch: 10.10.10.2
Router	Assumed Router: 10.100.152.1
IP Range	.1, .10, .20, .30
Broadcast Address	10.10.10.255

This subnet only communicates within itself. This subnet uses a multitude of management protocols. I have visualised its connections here:



Why do we assume router connections?:

There seemingly are numerous “floating” subnets. However the fact that they appear in the packet capture, implies that there is some connection to the rest of the network. As the device taking the capture can see all the devices, there must be some cabled route between all the subnets. Therefore I have had to assume the router connections to these subnets. In reality there may be other inactive routers that haven’t appeared in the capture.

1.6 Servers and Routers

To formally check off assessment objective 3, here is a list of all network devices and servers identified so far, in a complete list:

Networking Devices:

Address	Type
192.168.89.1	Router
192.168.88.1	Router
10.100.152.1	Router
192.168.143.1	Virtual (VM) Router
10.0.1.1	Offline/Missing Router
192.168.88.60	Switch
192.168.88.61	Switch
192.168.88.80	Switch
192.168.88.95	Switch

Table 13: Devices [1]

Servers:

Address	Type
192.168.88.130	Serial Device Server
192.168.57.2	SSH Server
10.100.152.10	DHCP Server
10.100.152.128	VPN Server

Table 14: Devices [2]

1.7 Holistic Topology

Mapping out the network as documented is rather difficult. To aide in this, there are 3 diagrams included, each representing a different interpretation of the network. These diagrams represent the following, in order:

Diagram Number	Purpose
1	A diagram showing all proved physical connections, these have direct proof through packets seen in WireShark. This includes the aforementioned “floating” subnets and is therefore partially inaccurate, however the generalised connections that are present are guaranteed to be factual.
2	A diagram showing all supposed connections, adding the router mentioned at 10.0.1.1 and the virtual router located at 192.168.143.0/24.
3	A diagram showing all assumed connections, as documented in the previous sections, connecting the whole network.

Table 15: Graphs

This final diagram will then be used to construct a rendition of the network in Cisco Packet Tracer.

1.7 Holistic Topology

Diagram 1:

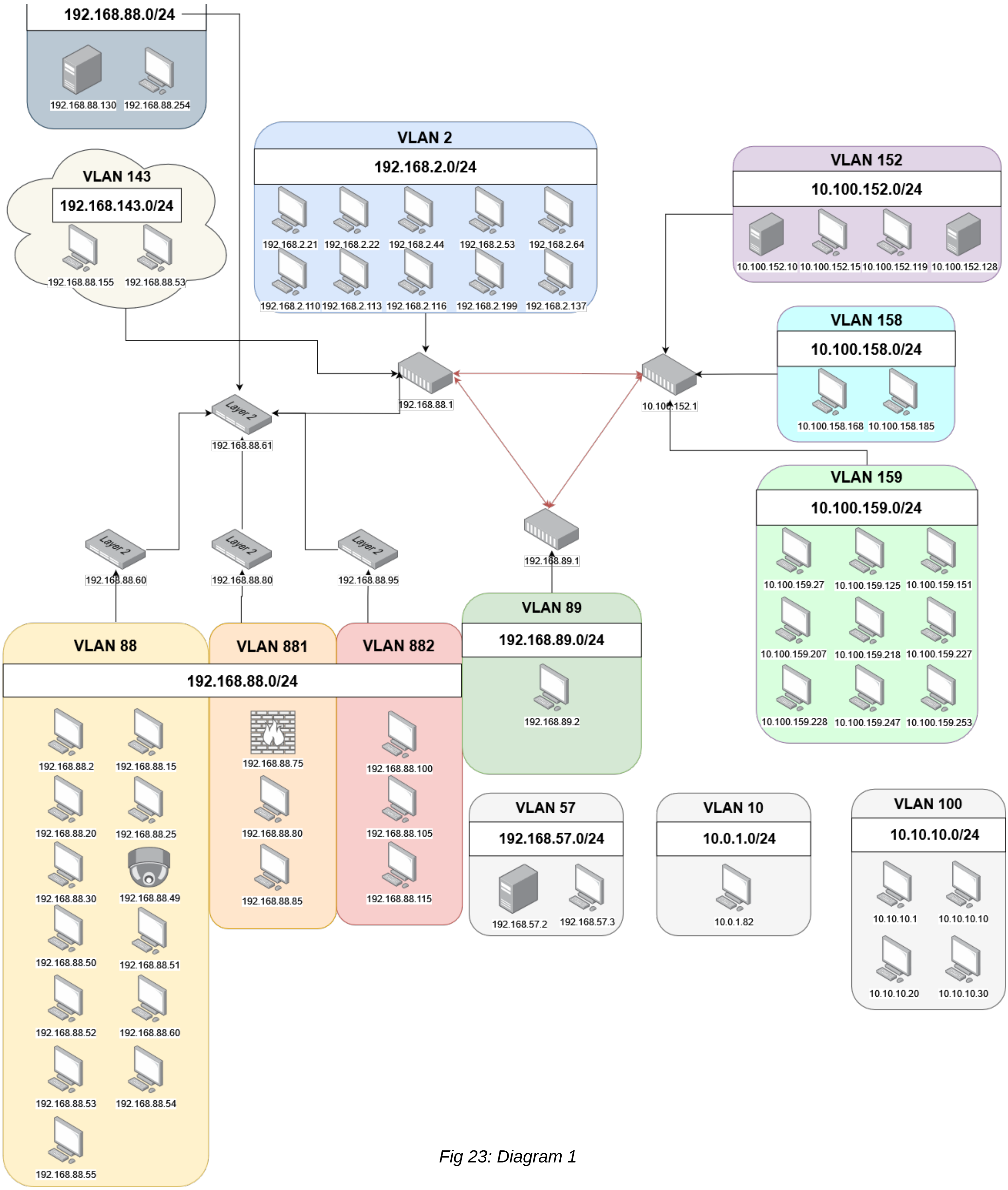


Fig 23: Diagram 1

1.7 Holistic Topology

Diagram 2:

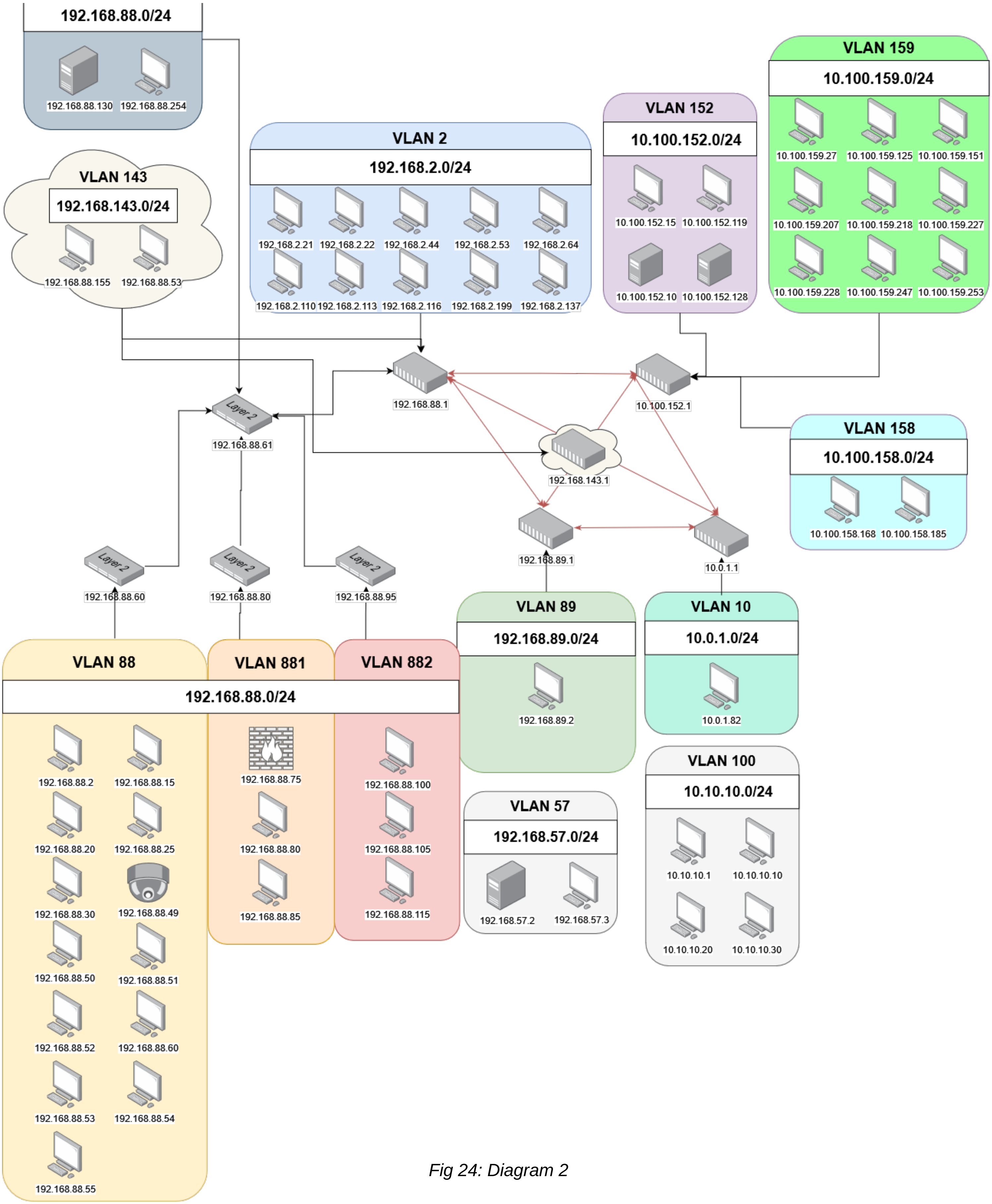


Fig 24: Diagram 2

1.7 Holistic Topology

Diagram 3: Red switches represent assumed devices, they may not appear in the capture due to the layer the scan occurred on.

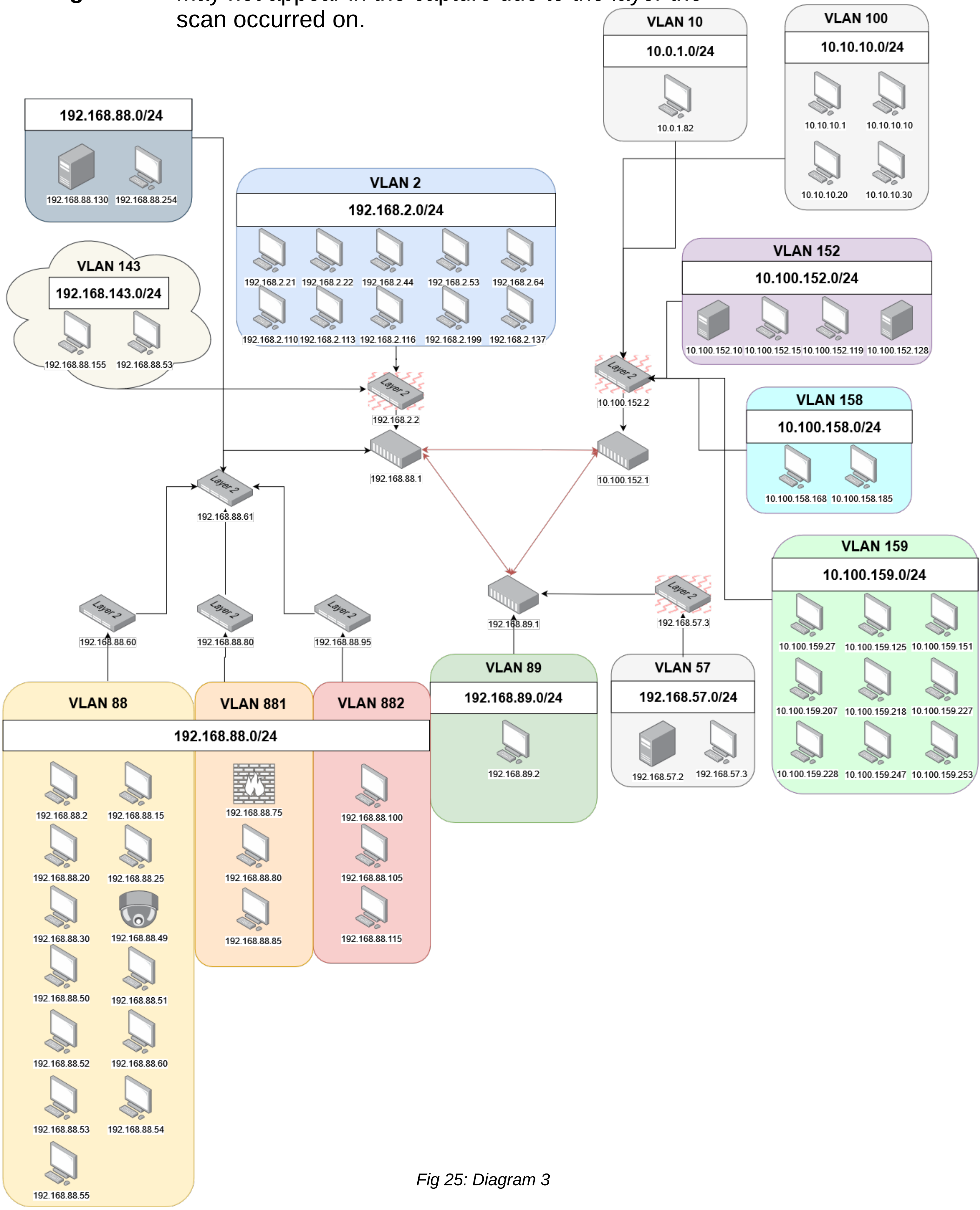


Fig 25: Diagram 3

1.8 Modelling Topology in Cisco Packet Tracer

Overview:

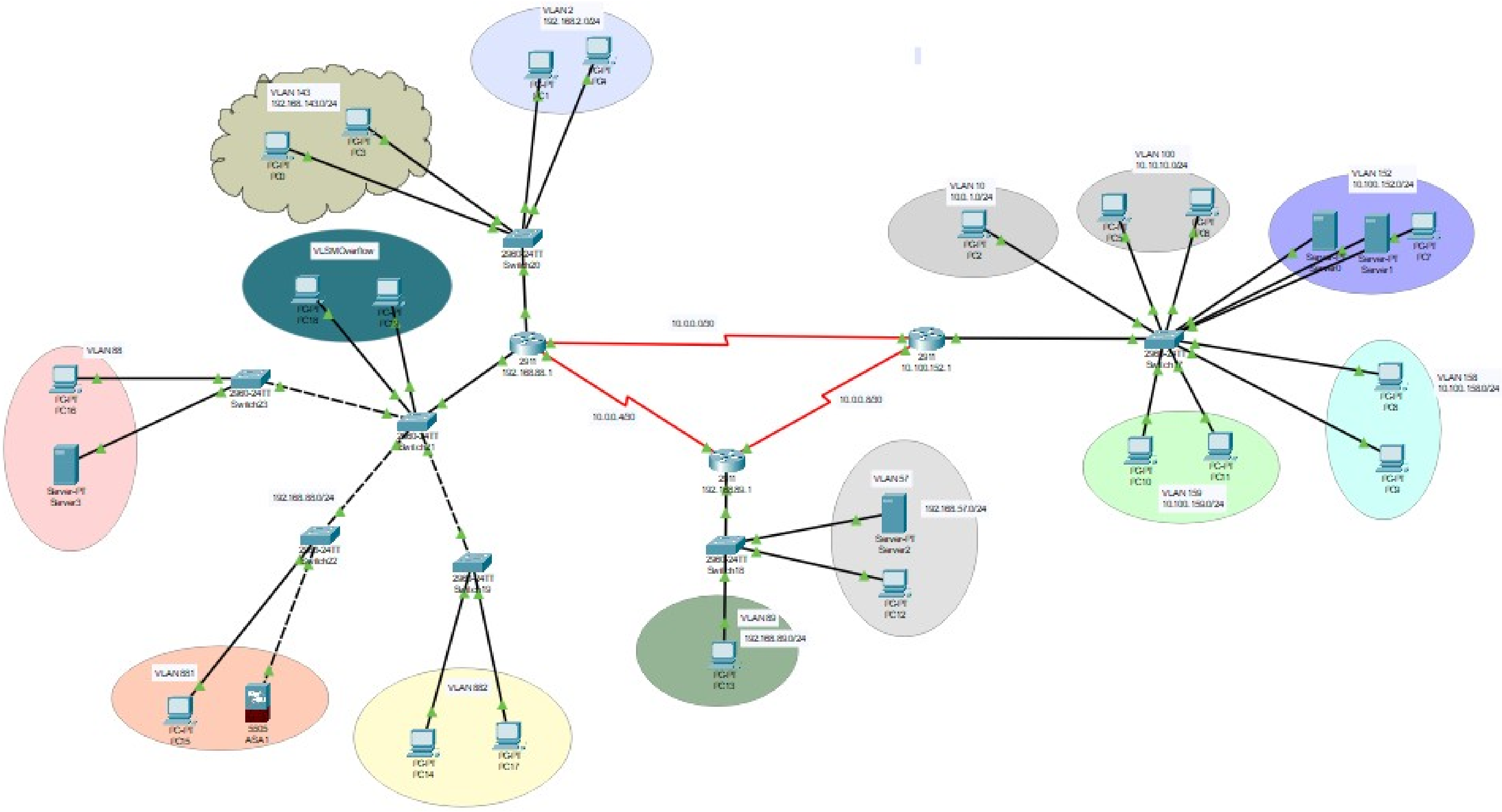


Fig 26: Topology 1

First devices were placed like the figure above, akin to topology 3, then the following configuration commands were run.

Switch configuration:

```
Switch(config-if-range)#int range fa0/2-3
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 100
% Access VLAN does not exist. Creating vlan 100
```

Fig 27: Making Vlans

Figure 27 assigns VLAN numbers to specific interfaces, repeat this on every switch for each connected VLAN.

```
Switch(config)#int range fa0/11
Switch(config-if-range)#switchport mode trunk

Switch(config-if-range)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to up

Switch(config-if-range)#do wr
```

Fig 28: Trunking Switches

Figure 28 configures specific interfaces to be trunked, allowing for handling of multiple VLANs on a device, needed for inter VLAN routing.

```
Switch(config)#int range fa0/1-4
Switch(config-if-range)#switchport mode trunk
Switch(config-if-range)#switchport trunk allowed vlan 88,881,882
```

Fig 29: Trunking the big switch

To configure switch 192.168.88.61 (the top switch). The commands were needed to allow for trunking, down the switch chain to specific VLANs.

```
Switch(config)#vlan 88
Switch(config-vlan)#name "vlan88"
```

Fig 29: Assigning VLAN names

Additionally names must be assigned on both bottom and top switches, to allow for identification at different switch layers, this was not needed for the other VLANs contained on other switches.

Router configuration:

As for the router, many more commands were run:

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int se0/3/0
Router(config-if)#no sh

%LINK-5-CHANGED: Interface Serial0/3/0, changed state to down
```

Fig 30: Configure router to router

In Figure 30, we configure the inter-router connections. This network is assumed as it doesn't seem to appear in the capture. This is repeated on each router.

```
Router(config-if)#int gig 0/0
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up
```

Fig 31: Router to Router Interface

In Figure 31, we startup the connection linking router to switch, this is repeated for each router switch pairing.

```
Router(config)#int se 0/3/0
Router(config-if)#clock rate 64000
```

Fig 32: Set Interface Clock

Set clock rate for router to router serial DCE connection Set for each “clock interface”, 64000 is standard.

```
Router(config)#int se0/3/0
Router(config-if)#ip address 10.0.0.1 255.255.255.252
```

Fig 33: Assigning Private IP Range

This assigns an assumed private IP range used for inter-router communication to the relevant serial interface, repeated for all routers allowing for communication.

```
Router(config-subif)#int gig0/0.100
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0.100, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0.100, changed state to
up

Router(config-subif)#encapsulation dot1Q 100
Router(config-subif)#ip address 10.10.10.1 255.255.255.0
```

Fig 34: Encapsulation

Figure 34 show commands that select an interface, create a sub interface corresponding to the VLANs name, encapsulate with this name, and assign the VLANs network address. This is run on every router for every attached VLAN.

```
Router(config)#ip dhcp pool vlan152
Router(dhcp-config)#network 10.100.152.0 255.255.255.0
Router(dhcp-config)#default-router 10.100.152.1
Router(dhcp-config)#dns-server 10.100.152.1
Router(dhcp-config)#ex
```

Fig 35: Setting DHCP and DNS

```
Router(config)#ip dhcp pool 88
Router(dhcp-config)#network 192.168.88.0 255.255.255.192
Router(dhcp-config)#default-rout
Router(dhcp-config)#default-router 192.168.88.1
Router(dhcp-config)#dns-ser
Router(dhcp-config)#dns-server 192.168.88.1
Router(dhcp-config)#ex
Router(config)#ip dhcp pool 881
Router(dhcp-config)#network 192.168.88.64 255.255.255.224
Router(dhcp-config)#defa
Router(dhcp-config)#default-router 192.168.88.65
Router(dhcp-config)#dns-ser
Router(dhcp-config)#dns-server 192.168.88.65
Router(dhcp-config)#ex
Router(config)#ip dhcp pool 882
Router(dhcp-config)#netwo
Router(dhcp-config)#network 192.168.88.97 255.255.255.192
Router(dhcp-config)#network 192.168.88.96 255.255.255.192
Router(dhcp-config)#default-router 192.168.88.97
Router(dhcp-config)#dn
Router(dhcp-config)#dns-server 192.168.88.97
```

Fig 36: Full DHCP and DNS

We did discover a DHCP server, it sends very little traffic therefore it is likely that DHCP was performed on the switches. We configure that here in Figure 36.
This is the full DHCP enabling for the 192.168.88.0/24 subnet

```
Router(config)#router ospf 10
Router(config-router)#network 10.100.152.0 255.255.255.0 area 0
Router(config-router)#network 10.100.158.0 255.255.255.0 area 0
Router(config-router)#network 10.100.159.0 255.255.255.0 area 0
Router(config-router)#network 10.10.10.0 255.255.255.0 area 0
Router(config-router)#network 10.0.1.0 255.255.255.0 area 0
```

Fig 37: Configure OSPF

```
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.159.1      0   FULL/  -        00:00:35    10.0.0.2        Serial0/3/0
192.168.89.1      0   FULL/  -        00:00:39    10.0.0.6        Serial0/3/1
```

Fig 38: Debug OSPF

Figure 36 & 37 show the configuration of OSPF on our router. This allows for the advertisement of routes around the network, allowing for routing between the routers not just inter-vlan.

Proof of configuration:

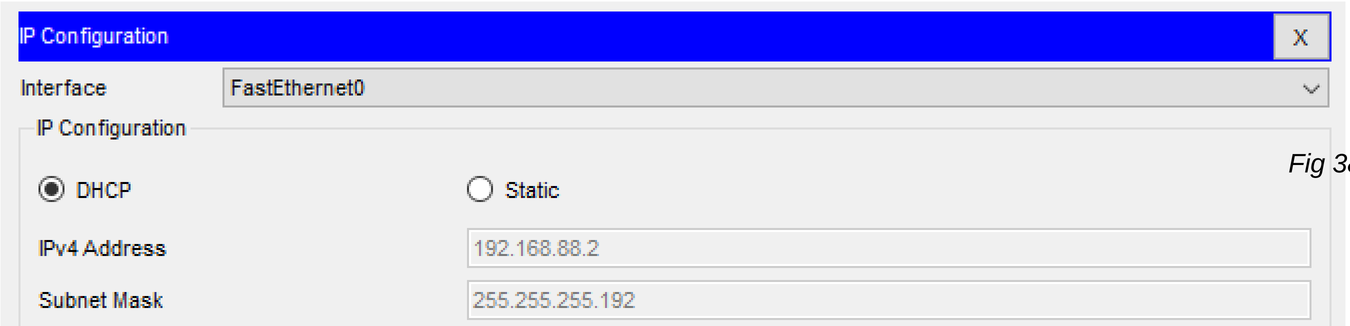


Fig 38: DHCP Proof

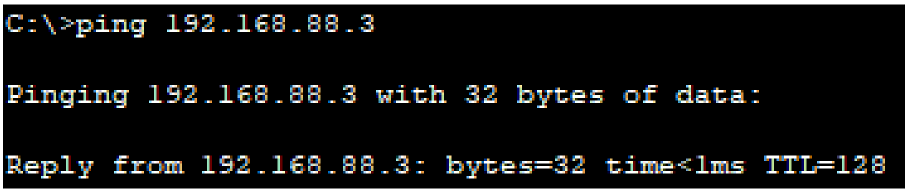


Fig 39: Intra-VLAN routing

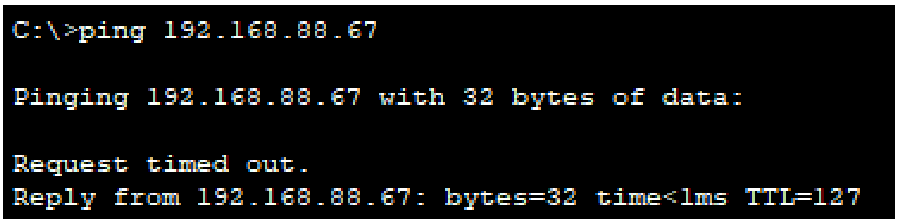


Fig 40: Inter VLAN Routing

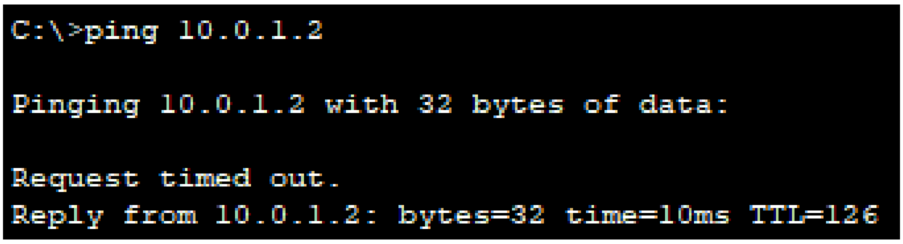


Fig 41: OSPF Full Functionality

Figure Number	Explanation
Fig 38	Here is the automatic IP allocation of the 192.168.88.2 endpoint, to the valid VLAN, performed by DHCP, showing that its operational.
Fig 39	This shows a successful ping, from our device on 192.168.88.2 to a device on the same VLAN, 192.168.88.3, showing that intra-VLAN routing is operational.
Fig 40	This shows a successful ping from our device to a device on a different VLAN configured on the same router, 192.168.88.67, showing inter-VLAN operations.
Fig 41	This shows a successful ping to a VLAN, configured on a different router, showing that we have full OSPF network connectivity.

Table 16: Proof[1]

2. Security:

2.2 Implementing Security Features

To help secure the system the following systems will be implemented:

Feature	Function
Enable SSH on routers	Allows for remote security updates and assessments, off-site audit log analysis.
Secure SSH on routers	Secure the SSH logins with secure RSA passwords.
Secure command line password on switches & routers	Add secure passwords for CLI access.
Secure enable on switches & routers	Add secure passwords to the enable command in the CLI.
Enable port security	Limit the number of MAC address on switch ports, preventing unauthorised network access.
Configure Access Control Lists (ACL)	Filter and restrict traffic types to designated network sections.
Implement DHCP snooping	Prevents DHCP spoofing attacks.
Zones of Trust	Add zones of trust between VLANs

Table 17: Security Upgrades

Enable and Secure SSH:

```
Router(config)#hostname 192.168.88.1
192.168.88.1(config)#ip domain-
192.168.88.1(config)#ip domain-name internal.network
192.168.88.1(config)#username admin secret strong_password
192.168.88.1(config)#crypto key generate rsa general-keys modulus 2048
The name for the keys will be: 192.168.88.1.internal.network

% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be non-exportable...[OK]
*Mar 1 0:5:40.603: %SSH-5-ENABLED: SSH 1.99 has been enabled
192.168.88.1(config)#ip ssh version 2
192.168.88.1(config)#line vty 0 15
192.168.88.1(config-line)#transport input ssh
192.168.88.1(config-line)#login local
192.168.88.1(config-line)#exit
192.168.88.1(config)#ip ssh time-out 60
192.168.88.1(config)#ip ssh authentication-retries 2
```

Fig 42: Secure SSH Config

Here is a breakdown for each command used to create a secure SSH connection. These commands have been adapted and run on every router.

Command	Function
ip domain-name internal.network	Creates a basic SSH service on the device located at 192.168.88.1 on the hostname internal.network.
username admin secret strong_password	Creates a login using default credentials to aide in testing.
crypto key generate rsa general-keys modulus 2048	Creates a 2048 bit RSA key for cryptographically secure SSH communication
ip ssh version 2	Update SSH to version 2, providing increased security over default version 1
line vty 0 15 transport input ssh login local exit	Configures the vty lines to only accept incoming SSH connections, stopping connections over insecure protocols, e.g.Telnet.
ip ssh time-out 60 ip ssh authentication-retries 2	Provides security through limiting password retries and forcing a timeout of 60s should the user get the password wrong twice.

Table 18: SSH Command steps

Secure the command line:

```
Switch(config)#hostname 192.168.88.61
192.168.88.61(config)#line console 0
192.168.88.61(config-line)#password console
192.168.88.61(config-line)#login
192.168.88.61(config-line)#exit
192.168.88.61(config)#enable password enable
192.168.88.61(config)#exit
```

Fig 43: Secure CLI Config

Here is a breakdown for each command used to create a secure command line interface (CLI). These commands have been adapted and run on every router and switch on the network.

Command	Function
Line console 0	Selects the console line
Password console	Sets the password for the console to “console”. Insecure set for testing.
Enable password enable	Sets the password for the elevated user account to “enable”. Insecure set for testing.

Enable Port Security:

```
192.168.88.61(config)#int range fa0/3
192.168.88.61(config-if-range)#switch
192.168.88.61(config-if-range)#switchport port
192.168.88.61(config-if-range)#switchport port-security
192.168.88.61(config-if-range)#swit
192.168.88.61(config-if-range)#switchport port-sec
192.168.88.61(config-if-range)#switchport port-security maximum 50
192.168.88.61(config-if-range)#switchport port-security violation restrict
192.168.88.61(config-if-range)#switchport port-security mac-address sticky
```

Fig 44: Port Security Config

Here is a breakdown of the commands run and its overall function. This has been enabled on every switch to every VLAN.

Command	Function
switchport port-security maximum 50	Sets the maximum MAC addresses per VLAN at 50
switchport port-security violation restrict	Restricts VLAN membership if over the maximum number
switchport port-security mac-address sticky	Sets a rule so that MAC addresses are remembered on specific ports. Therefore non recognised devices are unable to connect over the specified port. Treats all joined MAC addresses as manually allocated.
Overall Function	This prevents unauthorised access to VLANs through blocking the maximum number of unique devices able to be connected. This limit can change to fit the exact number of devices required. Additionally should an attacker attempt to unplug a device and join from the same port, it will be blocked unless it shares a MAC address, as MAC addresses are remembered on specific ports.

Table 19: CLI Command steps

Configure Access Control Lists:

Range	Allowed Protocols	Justification
General Function	-	We can control the traffic that can be sent around the network by implementing rules that manage communication. Managing this is important to security, as not all traffic is needed to be sent round the network. This is evident in the capture as traffic between routers is minimal.
Intra-VLAN	All Protocols	In VLANs its important to allow all traffic as to allow devices to engage in the full range of protocols available.
Inter-VLAN	SSH, ICMP, FTP, DNS, DHCP	In between the VLANs we can allow remote access for most needs, remote file download as VLANs may need to share work. Additionally DHCP is needed to resolve IPs on the router.
Inter-Router	SSH, ICMP	In between the routers, only SSH and ICMP are allowed, ICMP to check the status of remote hosts and SSH to allow for remote administrator control. This allows for remote access if needed without weakening network security.

Table 20: ACL List

Range	Commands	Function
Intra-VLAN	No commands needed	To allow all protocols no special configuration is required.

Table 21: Intra-VLN Table

Configure [INTER-VLAN] Access Control Lists:

```
192.168.88.1(config)#ip access-list extended INTER-VLAN
192.168.88.1(config-ext-nacl)#permit udp any any eq bootps
192.168.88.1(config-ext-nacl)#permit udp any any eq bootpc
192.168.88.1(config-ext-nacl)#permit tcp any any eq 22
192.168.88.1(config-ext-nacl)#permit icmp any any
192.168.88.1(config-ext-nacl)#permit tcp any any eq 21
192.168.88.1(config-ext-nacl)#permit tcp any any eq 20
192.168.88.1(config-ext-nacl)#permit tcp any any eq domain
192.168.88.1(config-ext-nacl)#permit udp any any eq domain
192.168.88.1(config-ext-nacl)#deny ip any any
192.168.88.1(config-ext-nacl)#exit
```

Fig 45: ACL Config 1

Inter-VLAN		
	ip access-list extended INTER-VLAN	Creates an ACL called “INTER-VLAN” used for the aforementioned job of connecting VLANS on a router.
	permit udp any any eq bootps	! Permit DHCP requests from clients
	permit udp any any eq bootpc	! Permit DHCP responses to clients
	permit tcp any any eq 22	! Permit SSH
	permit icmp any any	! Permit ICMP
	permit tcp any any eq 21	! Permit FTP command
	permit tcp any any eq 20	! Permit FTP data
	permit tcp any any eq domain	! Permit DNS (TCP)
	permit udp any any eq domain	! Permit DNS (UDP)
	deny ip any any	! Deny all other traffic

Table 22: INTER-VLN Breakdown

Configure this same ACL rule set on every router.

```
192.168.88.1(config)#int gig0/1.2
192.168.88.1(config-subif)#ip access-group INTER-VLAN in
192.168.88.1(config-subif)#int gig0/1.143
192.168.88.1(config-subif)#ip access-group INTER-VLAN in
```

Fig 46: ACL Assignment 1

Run this on every sub interface connected to a VLAN, swapping out, substitute .143 for your VLAN of choice. This adds the sub interface to the new ACL.

Configure [OUTBOUND] Access Control Lists:

```
192.168.88.1(config)#ip access-list extended OUTBOUND
192.168.88.1(config-ext-nacl)#permit icmp any any
192.168.88.1(config-ext-nacl)#permit tcp any any eq 22
192.168.88.1(config-ext-nacl)#deny ip any any
```

Fig 47: ACL Config 2

Inter-Router		
	permit icmp any any	! Permit ICMP
	permit tcp any any eq 22	! Permit SSH
	deny ip any any	! Deny all other traffic

Table 23: OUTBOUND Breakdown

Configure this same ACL rule set on every router.

```
192.168.88.1(config-if)#int se0/3/1
192.168.88.1(config-if)#ip access-group OUTBOUND out
```

Fig 48: ACL Assignment 2

Run this on every serial DCE (router to router) interface, subbing in the relevant interface value

Configuring ACLs like this allows for the following **Zones of Trust:**

Name	Trust Level	Justification
Intra-VLAN	High Trust	Allowed to send any data of any type, full freedom with communication, has access to non password protected data streams.
Inter-VLAN	Medium Trust	Allowed to send some useful data, and allowed anonymous access to computer’s file system via anonymous FTP login. Additionally has DNS capabilities.
Outbound	Low Trust	All streams of communication are password protected (SSH) and this level is only configured for remote administration. Incredibly limited by protocol, can only interact with machines that have SSH enabled.

Table 24: Zones of Trust

2.3 Evidence of Security Features

Proof of configuration:

Enable and Secure SSH:

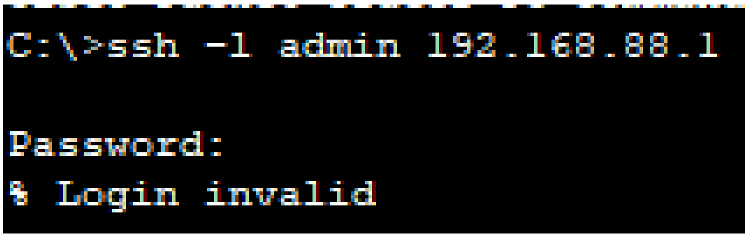


Fig 49: SSH Fail Login

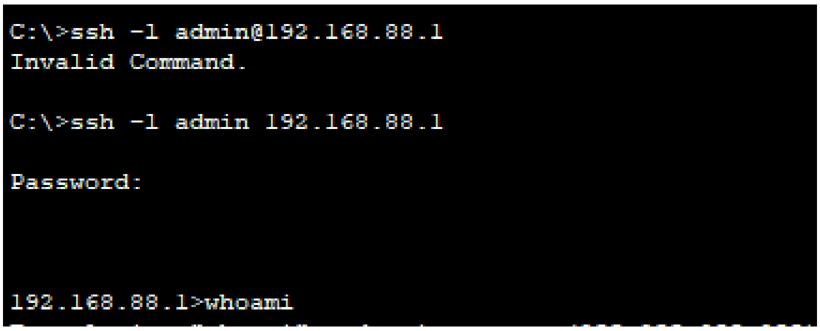


Fig 50: SSH Login Win

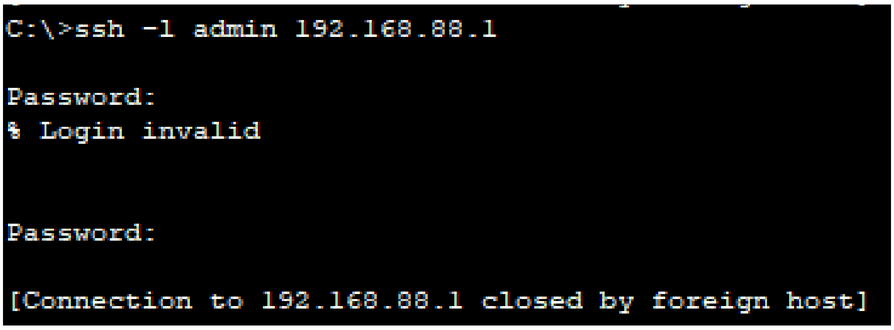


Fig 51: Double Fail

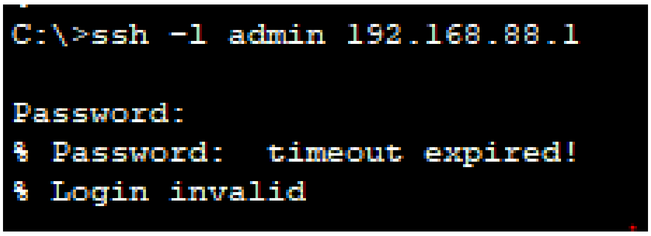


Fig 52: Timeout

Figure Number	Explanation
Fig 49	This figure features a failed login attempt from 192.168.0.2 to our router at 192.168.88.1. It failed as the wrong password was provided.
Fig 50	This shows a successful login as this command is entered again and the correct password is given. We now have the CLI of the target machine.
Fig 51	This shows the connection being closed automatically given two wrong password attempts.
Fig 52	This shows the connection closing due to the timeout expiring, set at 60 seconds.

Table 25: SecProof [1]

Proof of configuration:

Secure CLI & Enable Passwords:

```
User Access Verification

Password:
Password:
```

Fig 53: CLI Lockout

```
User Access Verification

Password:
Password:

192.168.88.61>
```

Fig 54: Successful login

```
192.168.88.61>en
Password:
Password:
```

Fig 55: Enable lockout

```
192.168.88.61>en
Password:
Password:
192.168.88.61#
```

Fig 56: Enable login

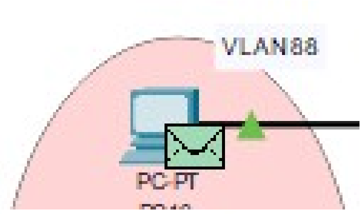
Figure Number	Explanation
Fig 53	This shows an attempted login to the CLI of a switch @ 192.168.88.61. The password was entered wrong therefore it was re-prompted.
Fig 54	After a successful login, as seen here, we have access to the CLI.
Fig 55	Attempting to access the “enabled” mode, with elevated privilege, another password is required, this is again failed and re-prompted.
Fig 56	After getting the enabled password correct we now have administrative privilege on the system.

Table 26: SecProof [2]

Proof of configuration:

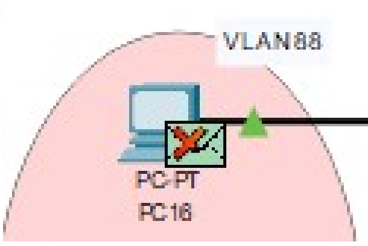
Access Control Lists [INTER-VLAN]:

To prove configuration lets find one example of the rules working to allow communication, and one where the rules work to block communication.




```
C:\>telnet 192.168.143.1
Trying 192.168.143.1 ...
```

Fig 57: Telnet Attempt [1]




```
C:\>telnet 192.168.143.1
Trying 192.168.143.1 ...
% Connection timed out; remote host not responding
```

Fig 58: Telnet Fail [1]

```
C:\>telnet 192.168.88.3
Trying 192.168.88.3 ...
% Connection refused by remote host
```

Fig 59: Telnet Example [1]]

```
C:\>ping 192.168.88.3

Pinging 192.168.88.3 with 32 bytes of data:

Reply from 192.168.88.3: bytes=32 time<1ms TTL=127
Reply from 192.168.88.3: bytes=32 time=9ms TTL=127
Reply from 192.168.88.3: bytes=32 time<1ms TTL=127
Reply from 192.168.88.3: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.88.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 9ms, Average = 2ms
```

Fig 60: Inter-VLAN Ping

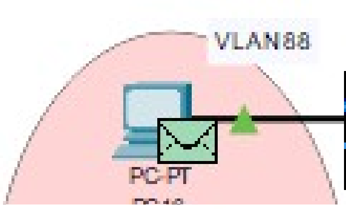
Figure Number	Explanation
Fig 57	Here the pc 192.168.88.2 is attempting to connect via telnet to a PC on a different VLAN (192.168.143.1), using an disallowed protocol.
Fig 58	Shown here the packet is dropped at its source, as it is unable to leave the VLAN, and the host returns as not reachable.
Fig 59	If it was contained in the VLAN, telnet would have been allowed returning this.
Fig 60	For evidence of functionality, we can simply use the ping command. This is because the ACL allows for ICMP packets. Therefore as a ping from 192.168.143.2 to 192.168.88.3 was successful, the ACL is configured correctly.

Table 27: SecProof [3]

Proof of configuration:

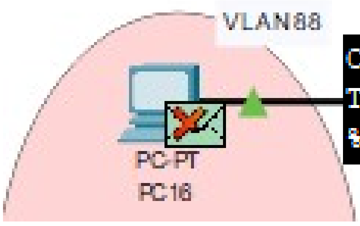
Access Control Lists [OUTBOUND]:

Much like with INTER-VLAN we need an example of a successful block and a successful transmission.




```
C:\>telnet 10.0.1.2
Trying 10.0.1.2 ...
```

Fig 61: Telnet Attempt [2]




```
C:\>telnet 10.0.1.2
Trying 10.0.1.2 ...
% Connection timed out; remote host not responding
```

Fig 62: Telnet Fail [2]

```
C:\>telnet 192.168.88.3
Trying 192.168.88.3 ...
% Connection refused by remote host
```

Fig 63: Telnet Example [2]]

```
C:\>ping 10.0.1.2

Pinging 10.0.1.2 with 32 bytes of data:

Request timed out.
Reply from 10.0.1.2: bytes=32 time=14ms TTL=126
Reply from 10.0.1.2: bytes=32 time=15ms TTL=126
Reply from 10.0.1.2: bytes=32 time=23ms TTL=126

Ping statistics for 10.0.1.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 23ms, Average = 17ms
```

Fig 64: Inter-Router Ping

Figure Number	Explanation
Fig 61	Here the pc 192.168.88.2 is attempting to connect via telnet to a PC on a different router (10.0.1.2), using an disallowed protocol.
Fig 62	Shown here the packet is dropped at its source, as it is unable to leave the VLAN, and the host returns as not reachable.
Fig 63	If it was contained in the VLAN, telnet would have been allowed, returning this:
Fig 64	For evidence of functionality, we can simply use the ping command. This is because the ACL allows for ICMP packets. Therefore as a ping from 192.168.143.2 to 10.0.1.2, a device on a different router, was successful, the ACL is configured correctly.

Table 28: SecProof [4]

Proof of configuration:

DHCP Snooping:

```
192.168.88.60(config)#ip dhcp snooping
192.168.88.60(config)#ip dhcp snooping vlan 88
192.168.88.60(config)#int fa0/3
192.168.88.60(config-if)#ip dhcp snooping trust
```

Fig 65: DHCP Snoop

The principle behind DHCP snooping is the validation of untrusted DHCP, passed from untrusted interfaces. This mitigation eliminates the threat of malicious actors posing as DHCP servers, intentionally distributing false IPs. As you can see in Fig 65 I attempted to integrate DHCP snooping into the network, however it seemed to always break the VLSM sub-netting within 192.168.88.0/24, I am unsure why, I therefore had to leave it out of the final configuration.

Port Security:

```
192.168.88.61#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
               (Count)       (Count)          (Count)
--------------------------------------------------------------------
        Fa0/1        50           2                  0         Restrict
        Fa0/2        50           2                  0         Restrict
        Fa0/3        50           3                  0         Restrict
----------------------------------------------------------------------
```

Fig 66: Port Security

Fig 66 shows port security configured correctly on our major 192.168.88.61 switch, with the correct address split and security action to follow best practices. This prevents any devices from hijacking a port, to masquerade as a privileged client endpoint, bolstering security substantially.