



PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David
SWANSEA - ABERTAWE

Applied Computing

Wales Institute of Science and Art (WISA)

Technical Project (ACCA7053)

Theme: Cyber Sleuth

Unmasking the Invisible Network Shadows

ETHICAL STATEMENT

The tools and techniques you are exposed to' in this lab is to work only on a virtual machine environment and the purpose is entirely to demonstrate various types of cyber-attacks. Experimentation with these tools, techniques, and resources is at the discretion of the instructor and institution. If you as a student consider using attack tools for educational purposes on an outside environment, you should get approval from your lecturer prior to any experimentation.

Note: Unauthorized access to data, computer, and network systems is a crime in many jurisdictions and often is accompanied by severe consequences, regardless of the perpetrator's motivations. It is the learner's responsibility, as the user of this material, to be cognizant of and compliant with computer use laws.

I - Getting Ready



Objectives

Note: All the Activities listed in this workbook is a continuous working manual. You may complete this during the course duration. The activities include Information Gathering, and Scanning the Network for Vulnerabilities. The objectives set to this activity need to be achieved overall by completing all the activities and demonstrating them during the scheduled time.

1. **Import and Start VMs:** Import the provided virtual machines into your virtualization platform. Power on the machines.
2. **Verify Network Configuration:** Perform the following network tests to confirm the VMs are correctly configured as per the provided network topology. Record the results for submission.
3. **Address Issues:** If the test results deviate from the expected outputs, upload the evidence to Moodle and consult your lecturer immediately.
4. **Setup Security-focused Operating System of your choice**
 - Add (For eg: Kali Linux) a security-focused OS to simulate assess to preform security measure in the network.
 - Assign a network address of your choice to the Security-focused OS(Kali Linux) to represent internal access in Swansea Network.
 - Connecting within Swansea Network should give you access to both Swansea and London Network.

Optional Changes: Considering the number of Virtual Machines, you might ignore ClientL1 in your Activities. For Testing Purposes from a Client Machine, use ClientS1. However, if you wish to include ClientL1 in your Topology/ Testing, you are more than welcome.

To import the given simulated network (OVA File) and test the configurations, troubleshoot the machines within the network ensuring the network is configured as stated.

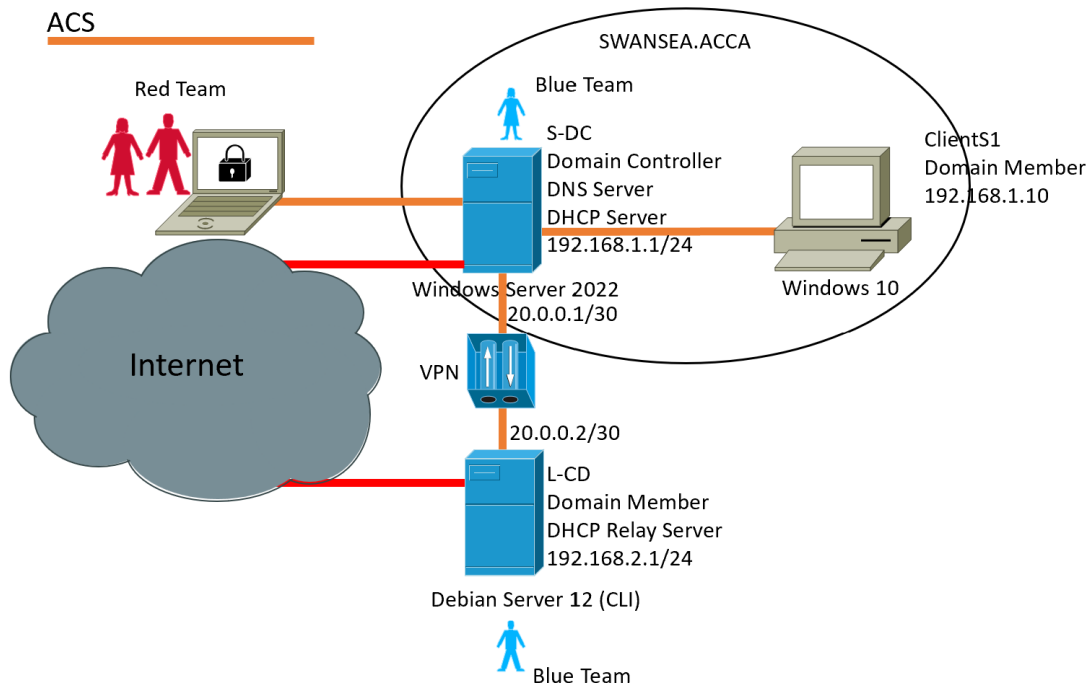
To setup OS to assess the impact of insecure configurations within the given network.

- ✓ **consider Information Security Elements, assessing risks to identify threats and vulnerabilities within the simulated network**

Upcoming Activities: You will be continuing to test the network, assess the security posture and enhance the network security and making the network Cyber risk resilient.

Part 1: Test and Troubleshoot (if needed) the Network as per the topology
Part 2: Setup Security OS within the Network

Lab Topology



- The OVA File is shared with you by your Lecturer.

Part 1: Basic Settings

1. Power-On the Virtual Machines and Test their Connectivity

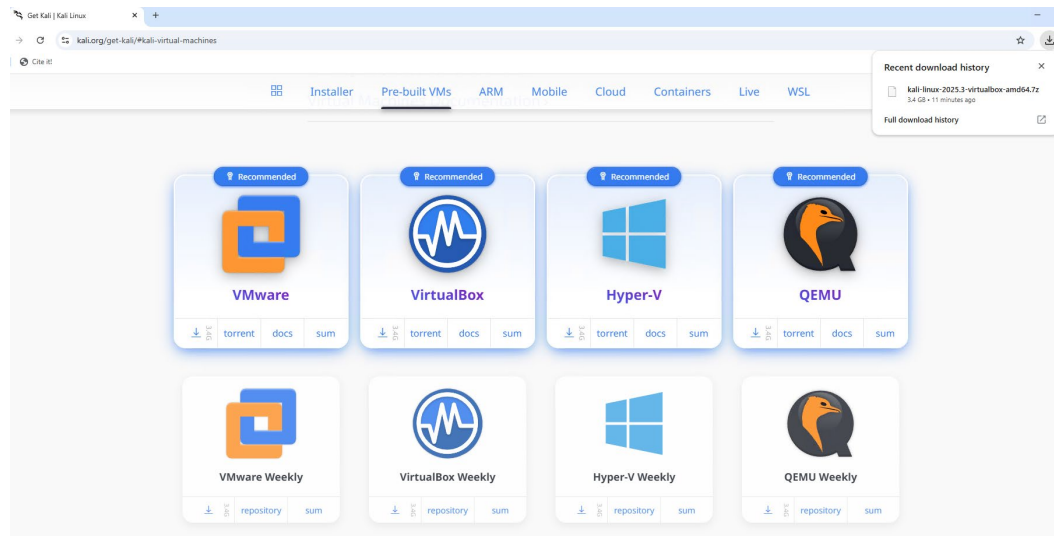
Useful Commands

- a. cmd – command prompt
 - b. ipconfig / ip address
 - c. ping “ip address”
 - d. ncpa.cpl – Network Adapters
 - e. route print
2. Capture a snapshot of the current VM state to back up your configurations using “Snapshot”.

Important: Submit the screenshot in Moodle – as part of your Coursework deliverables

Part 2: Install Security OS

1. In this example, Kali Linux is used for description. However, you are open to use any Security related OS.
2. Download Kali Linux for Virtual Box



3. Install Kali Linux and assign Swansea Network IP Address
4. Test Connectivity

Reference:

https://www.youtube.com/playlist?list=PLddfEpDxAoTt8mx4VcThYm_hSDid1WjZd

II - Information Gathering



1. **Understand multiple ways to Gather Information both Actively and Passively.**
2. **Scan your Network using Tools to identify the Services and its possible vulnerabilities.**

Optional Changes: Considering the number of Virtual Machines, you might ignore ClientL1 in your Activities. For Testing Purposes from a Client Machine, use ClientS1. However, if you wish to include ClientL1 in your Topology/ Testing, you are more than welcome.

PART 1: Recon a Network

Prior to start this activity, it is always recommended to capture a snapshot of the current VM state to back up your configurations using “Snapshot”.

Gathering Information on www.itsecgames.com

We will use `ping`, `tracert`, and `nslookup` (or `dig`) in Kali to passively gather details about a remote server's IP, network path, MTU, and DNS infrastructure, to understand attacker's reconnaissance for exploits like DoS or DNS-based attacks.

1. Basic Ping to Resolve IP and Test Connectivity

Why? Pinging reveals if the server is online, its IP address, and round-trip times (RTT), aiding in planning attacks like DDoS by confirming responsiveness.

Test ITSECGAMES.COM

```
C:\Users\Administrator>ping www.itsecgames.com

Pinging itsecgames.com [31.3.96.40] with 32 bytes of data:
Reply from 31.3.96.40: bytes=32 time=14ms TTL=255
Reply from 31.3.96.40: bytes=32 time=14ms TTL=255
Reply from 31.3.96.40: bytes=32 time=14ms TTL=255
Reply from 31.3.96.40: bytes=32 time=14ms TTL=255

Ping statistics for 31.3.96.40:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 14ms, Maximum = 14ms, Average = 14ms
```

```
(securityadmin@seclogin)-[~]
$ ping -c 4 www.itsecgames.com
PING itsecgames.com (31.3.96.40) 56(84) bytes of data.
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=1 ttl=255 time=14.8 ms
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=2 ttl=255 time=14.6 ms
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=3 ttl=255 time=14.6 ms
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=4 ttl=255 time=14.7 ms

--- itsecgames.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 14.577/14.682/14.828/0.092 ms
```

The `-c 4` flag limits to 4 pings.

ICMP Echo Requests (56 data bytes + 8 ICMP header = 64 total) get Echo Replies, showing IP (31.3.96.40), RTT, and 0% loss if reachable.

2. Discovering Maximum Transmission Unit (MTU) with Fragmentation Flags

Why? MTU knowledge helps craft oversized packets for fragmentation attacks or optimize payloads.

How? Use the `-M do` (Don't Fragment) flag with varying packet sizes.

```
C:\Users\Administrator>ping www.itsecgames.com -f -l 1400

Pinging itsecgames.com [31.3.96.40] with 1400 bytes of data:
Reply from 31.3.96.40: bytes=1400 time=14ms TTL=255
Reply from 31.3.96.40: bytes=1400 time=14ms TTL=255
Reply from 31.3.96.40: bytes=1400 time=15ms TTL=255
Reply from 31.3.96.40: bytes=1400 time=14ms TTL=255

Ping statistics for 31.3.96.40:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 14ms, Maximum = 15ms, Average = 14ms
```

-f: Sets the "Don't Fragment" (DF) bit.

-l 1400: Specifies 1400 bytes of data. This is the payload size.

Total Packet Size: 1400 (data)

```
(securityadmin@seclogin)-[~]
$ ping -M do -s 1400 -c 4 www.itsecgames.com
PING itsecgames.com (31.3.96.40) 1400(1428) bytes of data.
1408 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=1 ttl=255 time=17.7 ms
1408 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=2 ttl=255 time=14.7 ms
1408 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=3 ttl=255 time=14.9 ms
1408 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=4 ttl=255 time=16.9 ms

--- itsecgames.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 14.702/16.052/17.706/1.282 ms
```

```
(securityadmin@seclogin)-[~]
$ ping -M do -s 1473 -c 4 www.itsecgames.com
PING itsecgames.com (31.3.96.40) 1473(1501) bytes of data.
ping: sendmsg: Message too long
ping: sendmsg: Message too long
ping: sendmsg: Message too long
ping: sendmsg: Message too long

--- itsecgames.com ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3078ms
```

Linux uses **-M do -s 1400**.

-M do: Sets the "Don't Fragment" (DF) bit in the IP header, ensuring the packet is not fragmented if it exceeds the MTU.

-s 1472: Specifies 1472 bytes of data in the payload.

-c 4: Limits the ping to 4 packets.

Total Packet Size: 1472 (data) + 8 (ICMP header) + 20 (IP header) = 1500 bytes.

This command works on Linux systems with a ping version that supports the **-M** option (e.g., `iputils-ping`). If successful (0% packet loss), it indicates the MTU is at least 1500 bytes.

Reflection Thoughts:

On getting the outcome,

"1400-byte ping succeeds, while 1473-byte ping fails with "Frag needed and DF set."

What is the reason?

3. Tracing the Route with Traceroute and Understanding TTLs

Why? Maps the network path, exposing routers for potential attacks.

How?

```
C:\Users\soac-nk>tracert itsecgames.com

Tracing route to itsecgames.com [31.3.96.40]
over a maximum of 30 hops:

  1  <1 ms    <1 ms    <1 ms    172.29.1.254
  2  <1 ms    <1 ms    <1 ms    172.30.2.254
  3  <1 ms    <1 ms    <1 ms    172.30.200.2
  4  <1 ms    <1 ms    <1 ms    159.86.188.126
  5   1 ms    1 ms     1 ms     159.86.189.170
  6   2 ms    1 ms     1 ms     psba.bradss-sbr2.ja.net [146.97.41.6]
  7   5 ms    5 ms     5 ms     ae2.bradss-sbr2.ja.net [146.97.41.5]
  8   5 ms    5 ms     5 ms     ae27.londpg-sbr2.ja.net [146.97.33.33]
  9   5 ms    5 ms     5 ms     ae29.londhx-sbr1.ja.net [146.97.33.1]
 10   5 ms    5 ms     5 ms     ae0.londhx-ban2.ja.net [146.97.35.198]
 11   5 ms    5 ms     5 ms     ldn-b2-link.ip.twelve99.net [62.115.175.130]
 12   5 ms    5 ms     5 ms     ldn-bb2-link.ip.twelve99.net [62.115.120.238]
 13  12 ms    12 ms    12 ms     adm-bb2-link.ip.twelve99.net [62.115.137.235]
 14  10 ms    11 ms    10 ms     adm-b10-link.ip.twelve99.net [62.115.120.229]
 15  11 ms    11 ms    11 ms     195.90.97.126
 16  12 ms    12 ms    12 ms     195.90.97.171
 17  14 ms    13 ms    14 ms     ae20-r2-tb-abl01.transip.net [157.97.168.30]
 18  12 ms    13 ms    13 ms     80.69.64.5
 19  12 ms    12 ms    12 ms     178-18-84-1.colo.transip.net [178.18.84.1]
 20  14 ms    14 ms    14 ms     web.mmebvba.com [31.3.96.40]

Trace complete.
```

```
(securityadmin@seclogin)-[~]
$ traceroute itsecgames.com
traceroute to itsecgames.com (31.3.96.40), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  2.144 ms  1.862 ms  1.722 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
```

```
C:\Users\Administrator>tracert itsecgames.com

Tracing route to itsecgames.com [31.3.96.40]
over a maximum of 30 hops:

  1    14 ms    15 ms    14 ms    web.mmebvba.com [31.3.96.40]

Trace complete.

C:\Users\Administrator>tracert www.itsecgames.com

Tracing route to itsecgames.com [31.3.96.40]
over a maximum of 30 hops:

  1    14 ms    14 ms    14 ms    web.mmebvba.com [31.3.96.40]

Trace complete.
```

```
C:\Users\soac-nk>ping www.itsecgames.com -i 4 -n 1

Pinging itsecgames.com [31.3.96.40] with 32 bytes of data:
Reply from 159.86.188.126: TTL expired in transit.

Ping statistics for 31.3.96.40:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
```

```
(securityadmin@seclogin)-[~]
$ ping -t 4 -c 1 www.itsecgames.com
PING itsecgames.com (31.3.96.40) 56(84) bytes of data:
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=1 ttl=255 time=14.7 ms

--- itsecgames.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 14.719/14.719/14.719/0.000 ms
```

Uses TTL increments

each hop replies with "Time Exceeded" until the target responds.

Asterisks * indicate blocked probes.

Reflection Thoughts:

On getting the outcome using different options in Ping command,

can you find out more options that you can use with ping to further dive into the network to identify more details?

4. DNS Reconnaissance with Nslookup or Dig

Why? Reveals NS and IPs for DNS attacks (e.g., spoofing).

How?

```
C:\Users\Administrator>nslookup
Default Server:  UnKnown
Address:  192.168.1.1

> set type=a
> www.itsecgames.com
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:      itsecgames.com
Address:   31.3.96.40
Aliases:   www.itsecgames.com
```

```
> set type=cname
> www.itsecgames.com
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
www.itsecgames.com      canonical name = itsecgames.com
```

```
> set type=ns
> www.itsecgames.com
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
www.itsecgames.com      canonical name = itsecgames.com
itsecgames.com          nameserver = ns54.domaincontrol.com
itsecgames.com          nameserver = ns53.domaincontrol.com

ns54.domaincontrol.com  internet address = 173.201.74.27
ns54.domaincontrol.com  AAAA IPv6 address = 2603:5:22a2::1b
ns53.domaincontrol.com  internet address = 97.74.106.27
ns53.domaincontrol.com  AAAA IPv6 address = 2603:5:21a2::1b
> set type=a
> ns53.domaincontrol.com
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:      ns53.domaincontrol.com
Address:   97.74.106.27
```

In Linux

```
(securityadmin@seclogin)-[~]
$ dig itsecgames.com

; <<>> DiG 9.20.11-4+b1-Debian <<>> itsecgames.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2948
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:;, udp: 4000
;; QUESTION SECTION:
;itsecgames.com.                IN      A
;; ANSWER SECTION:
itsecgames.com.                15      IN      A      31.3.96.40

;; Query time: 8 msec
;; SERVER: 10.0.2.3#53(10.0.2.3) (UDP)
;; WHEN: Sun Oct 12 09:51:57 BST 2025
;; MSG SIZE rcvd: 59
```

```
(securityadmin@seclogin)-[~]
$ dig +short a itsecgames.com
31.3.96.40
```

```
(securityadmin@seclogin)-[~]
$ dig +short ns itsecgames.com
ns53.domaincontrol.com.
ns54.domaincontrol.com.
```

```
(securityadmin@seclogin)-[~]
$ dig +short a ns53.domaincontrol.com
97.74.106.27
```

```
(securityadmin@seclogin)-[~]
$ ping -c 4 ns53.domaincontrol.com
PING ns53.domaincontrol.com (97.74.106.27) 56(84) bytes of data.
64 bytes from ns53.domaincontrol.com (97.74.106.27): icmp_seq=1 ttl=255 time=14.7 ms
64 bytes from ns53.domaincontrol.com (97.74.106.27): icmp_seq=2 ttl=255 time=14.1 ms
^C
--- ns53.domaincontrol.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 14.111/14.413/14.715/0.302 ms
```

`dig` provides detailed DNS records.

`+short` simplifies output. Shows CNAME, A records, and NS IPs.

Pinging confirms reachability.

Reflection Thoughts:

What is so important in getting information like this?

Why do we get an output like this?

“SWANSEA.ACCA” - Network

PART 2: Reconnaissance on Server swansea.acca

Apply techniques to the SWANSEA.ACCA network from Kali, targeting S-DC and L-CD.

This tutorial assumes Kali is on 192.168.1.0/24 with 192.168.1.2 as IP Address and can route via 20.0.0.0/30.

What is the Objective?

Identify (Recon) any internal/external vulnerable services (DNS, DHCP, IIS) within the Network.

1. Basic Ping, MTU Discovery and Traceroute

Why?

Test connectivity to S-DC (swansea.acca)

MTU for potential exploits.

Note: The MTU could be used during your further test while scanning the network.

Map paths

Find information about routing

Hints:

ping -c 4 swansea.acca

ping -M do -s 1400 -c 4 swansea.acca

ping -M do -s 1473 -c 4 swansea.acca

ping -c 4 l-cd.swansea.acca

ping -t 2 -c 1 l-cd.swansea.acca

tracert swansea.acca

dig swansea.acca

III – Packet Captures



Introduction to Network Traffic Analysis

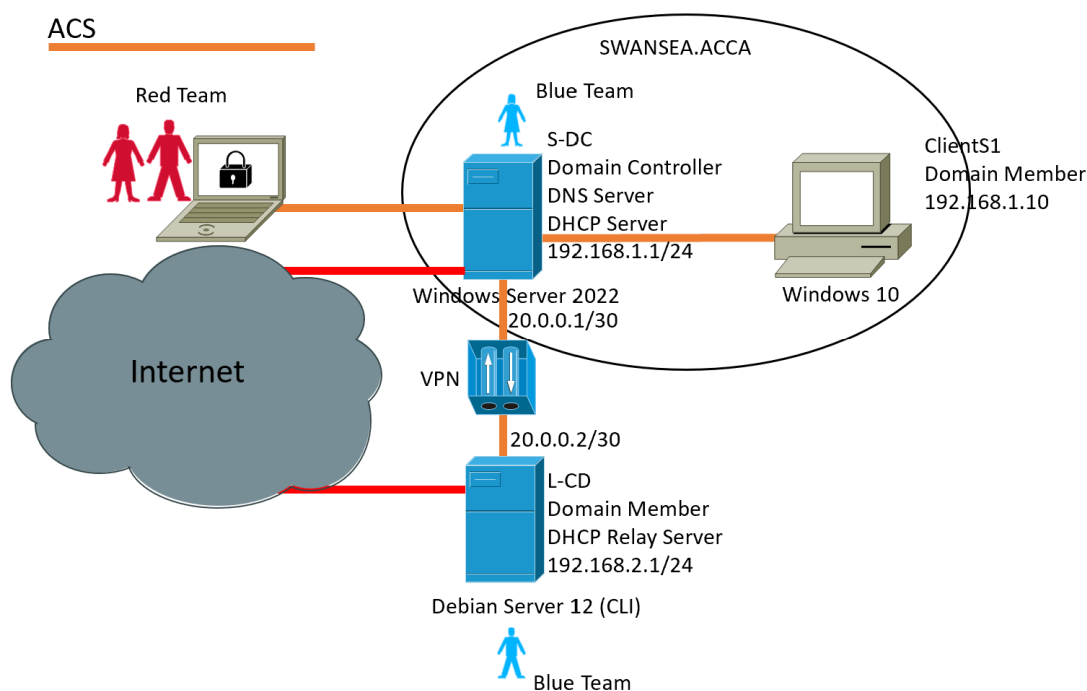
Wireshark, a powerful open-source packet analyzer used to capture and inspect network traffic. Wireshark helps you understand how data moves across networks by displaying packets in detail, including protocols, source/destination addresses, and payload information.

In this activity, you'll install Wireshark (or its command-line equivalent, tshark, for CLI environments) on three machines in the provided topology:

S-DC

Client S1

L-CD



After installation, you'll generate simple network traffic using commands like `ping`, `tracert` (or `traceroute` on Linux), `ftp`, and `telnet`. You'll capture this traffic on each machine and perform basic analysis. This will help you observe ICMP, TCP, and other protocols in action.

Ethical Considerations

- Capturing packets may reveal sensitive data; avoid capturing in production environments.
- FTP and Telnet transmit data in plain text—do not use real credentials.
- This activity focuses on observation; no actual exploitation is involved.

1. Installing Wireshark

Follow these steps to download and install Wireshark on each machine. Note that Windows machines will use the graphical Wireshark interface, while the Debian CLI server will use `tshark` (the text-based version included with Wireshark) since it lacks a GUI.

a. Install on S-DC (Windows Server 2022)

- Open a web browser (e.g., Microsoft Edge) and navigate to the official [Wireshark website](https://www.wireshark.org/download.html): `https://www.wireshark.org/download.html`.
- Under the "Stable Release" section, select the Windows Installer (64-bit) for the latest version. Download the `.exe` file (e.g., `Wireshark-win64-x.x.x.exe`).
- Locate the downloaded file in your Downloads folder and double-click it to run the installer.
- In the Setup Wizard: Click "Next" on the welcome screen, Accept the License Agreement and click "Next", Choose components: Ensure "Wireshark" and "Npcap" (the packet capture driver) are selected. Npcap is essential for capturing packets—install it if prompted.
- Select installation location (default is fine) and click "Next".
- Choose to install USBPcap if needed (optional for this activity).
- Click "Install". If User Account Control (UAC) prompts, approve it.
- During installation, the Npcap installer may appear—accept its license and install with default options. Once complete, click "Finish". Restart the server if prompted.

b. Verify installation:

- Search for "Wireshark" in the Start menu and launch it.
- You should see a list of network interfaces (e.g., Ethernet adapter connected to 192.168.1.0/24).

c. Install on Client S1 (Windows 10)

- Repeat the same steps as above for S-DC, using the same download link.

Key differences: Windows 10 may already have some drivers; ensure Npcap installs correctly.

- ii. Verify: Launch Wireshark and confirm interfaces (e.g., the one with IP 192.168.1.10).

d. Install on L-CD (Debian Server 12 CLI)

Since this is a CLI-only server, we'll install the Wireshark package, which includes `tshark`. No GUI is needed.

- i. Open the terminal
- ii. Update package lists: Run `sudo apt update` (enter root password if prompted).
- iii. Install Wireshark: Run `sudo apt install wireshark -y`. During installation, you'll be prompted to allow non-superusers to capture packets—select "Yes" for simplicity (this creates a `wireshark` group).
- iv. Add your user to the wireshark group: Run `sudo usermod -aG wireshark \$USER`. Log out and log back in for changes to take effect.
- v. Install `tshark` specifically if needed (it's included): Run `sudo apt install tshark -y`.
- vi. Verify: Run `tshark -v` to check the version (should show Wireshark 4.x or higher). Run `tshark -D` to list interfaces

2. Preparing for Traffic Generation

To generate FTP and Telnet traffic, set up temporary servers on S-DC (as the target). This allows connections from Client S1 or L-CD.

They are already installed and pre-configured. Make sure the services are on and running.

If you have accidentally uninstalled them, install and configure them.

a. Enable Telnet Server on S-DC (Windows Server 2022)

- 1. Open PowerShell as Administrator.
- 2. Run `Install-WindowsFeature Telnet-Server`.
- 3. Start the service: `Start-Service TlntSvr`.
- 4. Set to auto-start: `Set-Service TlntSvr -StartupType Automatic`.
- 5. Allow through firewall: `netsh advfirewall firewall add rule name="Telnet" dir=in action=allow protocol=TCP port=23`.

3. Capturing and Generating Traffic

Perform captures on each machine while generating traffic from others. Start with simple commands, capture, stop, and save the capture file (.pcap) for analysis.

a. General Capture Instructions

On Windows (S-DC and Client S1):

b. Launch Wireshark > Select the correct interface

On Debian (L-CD):

c. Run `tshark -i <interface> -w capture.pcap`` (e.g., `-i eth0`) to start capturing to a file. Press Ctrl+C to stop.

d. Filter during capture if needed (e.g., "icmp" for ping), but start without filters for full visibility.

e. Generate traffic from one machine to another (e.g., from Client S1 to S-DC).

f. Ping (ICMP Echo): Observe ICMP request/reply packets for connectivity testing.

i. On the capturing machine (e.g., S-DC), start capture.

ii. From Client S1: Open Command Prompt > Run `ping 192.168.1.1`` (S-DC's IP).

iii. From L-CD: Run `ping 192.168.1.1`` (assuming routing works).

iv. Stop capture after 4-5 pings.

v. Repeat captures on other machines.

g. Tracert/Traceroute (ICMP/UDP): See path discovery packets, including Time Exceeded messages.

i. Start capture on S-DC.

ii. From Client S1: `tracert 192.168.2.1`` (L-CD's IP).

iii. From L-CD: `traceroute 192.168.1.10`` (Client S1's IP).

iv. Stop and save capture.

h. FTP (File Transfer Protocol): Capture TCP connection, authentication, and data transfer.

i. Start capture on S-DC (the FTP server).

- ii. From Client S1: Open Command Prompt > ``ftp 192.168.1.1`` > Login with test credentials (e.g., anonymous) > Run ``dir`` to list files > ``bye`` to exit.
 - iii. From L-CD: Run ``ftp 192.168.1.1`` > Same steps.
 - iv. Stop capture.
- i. Telnet: Observe TCP session with plain-text commands.
 - i. Start capture on S-DC (the Telnet server).
 - ii. From Client S1: ``telnet 192.168.1.1`` > Login if prompted > Run simple commands like ``dir`` > Exit with ``logout``.
 - iii. From L-CD: ``telnet 192.168.1.1`` > Same.
 - iv. Stop capture.

****Tip:**** If connections fail, check firewalls and services. Capture failed attempts to see SYN/ACK or RST packets.

4. Analyzing Captured Packets

Load your .pcap files in Wireshark (on Windows) or use ``tshark -r capture.pcap`` on Debian for text output. For GUI analysis:

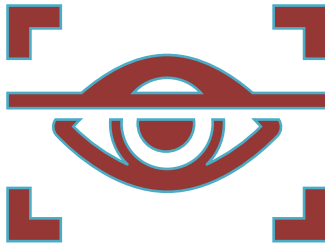
- a. Open the capture file in Wireshark.
- b. Apply filters:
 - i. ``icmp``: Shows ping requests/replies. Look at Type (8 for request, 0 for reply), source/destination IPs.
 - ii. ``ip.addr == 192.168.1.1``: Filter by IP.
 - iii. ``tcp.port == 21``: For FTP control channel. Expand packets to see commands like USER, PASS.
 - iv. ``tcp.port == 23``: For Telnet. View the Info column for commands; expand Telnet section to see plain-text data.
 - v. ``udp``: May appear in tracer (Windows uses ICMP, but some variants use UDP).
- c. Key things to observe:
 - i. Packet Structure: Ethernet frame > IP header > TCP/UDP/ICMP > Data.
 - ii. Three-Way Handshake (TCP): SYN, SYN-ACK, ACK for FTP/Telnet.
 - iii. Errors: Look for ICMP Destination Unreachable or TCP RST.
 - iv. Statistics: Go to Statistics > Conversations > IPv4 to see traffic summary.
 - v. Export: Save filtered packets or export objects (e.g., FTP data).

- d. On Debian with tshark: Run `tshark -r capture.pcap -Y "icmp" -V`` for detailed ICMP dissection.

Reflection Thoughts:

1. What differences did you notice between ICMP (ping) and TCP (FTP/Telnet) packets?
2. How does the topology (subnets, VPN) affect packet paths in tracer?
3. Why are FTP and Telnet insecure based on your captures?

III Scanning your Network



1. **Understand multiple ways to Gather Information both Actively and Passively.**
2. **Scan your Network using Tools to identify the Services and its possible vulnerabilities.**

PART 1: Scan a Network

Prior to start this activity, it is always recommended to capture a snapshot of the current VM state to back up your configurations using “Snapshot”.

“SWANSEA.ACCA” - Network

Scan and Enumerate the SWANSEA.ACCA network from Kali, targeting S-DC and L-CD.

This tutorial assumes Kali is on 192.168.1.0/24 with 192.168.1.2 as IP Address and can route via 20.0.0.0/30.

What is the Objective?

Scan and Identify any internal/external vulnerable services (DNS, DHCP, IIS) within the Network.

Here, we use NMAP. Refer to <https://nmap.org/book/port-scanning-options.html> for more information.

2. Scan the Network:

(Note: The Commands are given as Examples. Feel free to modify the commands based on your understanding)

Ping Scan

```
sudo nmap -sn 192.168.0.0/16
```

This is a ping scan (-sn) that disables port scanning and just checks if hosts respond to ICMP echo (ping) or other probes.

It's quick for discovering live hosts.

```
(securityadmin@seclogin)-[~]
$ sudo nmap -sn 192.168.0.0/16
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-19 20:05 BST
Nmap scan report for 192.168.1.1
Host is up (0.00044s latency).
MAC Address: 08:00:27:10:8C:B0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.1.2
Host is up.
Nmap scan report for 192.168.2.1
Host is up (0.0011s latency).
```

UDP Scan

```
sudo nmap -sU -p- 192.168.1.1
```

This is a udp scan Performs a UDP port scan to identify open UDP ports.

Version Detection

```
sudo nmap -sV -p 1-1024 192.168.1.1
```

Enables service version detection on specified ports.

OS Detection

```
sudo nmap -O 192.168.1.1
```

Performs OS detection by fingerprinting the target's operating system.

Aggressive Scan

```
sudo nmap -A 192.168.1.1
```

Aggressive scan combining OS detection, version scanning, script scanning, and traceroute.

Timing to Aggressive Level

```
sudo nmap -T4 -p- 192.168.1.1
```

Adjusts timing to aggressive level (T4) for faster scanning.

"full" or "connect" scan

(completes the entire TCP 3-way handshake to check port status. It's straightforward but not stealthy, as it mimics a real connection attempt)

```
sudo nmap -sT -p- 192.168.1.1 (repeat for other IPs).
```

"half-open" or "stealth" scan

it's the default Nmap TCP scan for root users. Nmap sends a SYN packet (as if starting a connection).

If the port is open, the target responds with SYN-ACK;

Nmap then sends RST to abort (no full handshake).

If closed, target sends RST. No response = filtered.

```
sudo nmap -sS -p- 192.168.1.1 (repeat for others).
```

XMAS Scan

An "inverse" or non-standard TCP flag scan

```
sudo nmap -sX -p- 192.168.1.1 (repeat for others).
```

On Linux (Debian), you might see more "open|filtered" results; on Windows, it may mark everything as closed.

Note differences on Windows vs. Debian. Why might this scan fail on certain OSes?

SYN/FIN Scan (TCP FIN Scan)

Another inverse flag scan using the FIN flag, often grouped with SYN but distinct—here, it refers to FIN scan, sometimes combined in discussions with SYN for contrast.

```
sudo nmap -sF -p- 192.168.1.1 (repeat for others).
```

Run and compare to XMAS. How does the single FIN flag change results vs. multiple flags?

ACK Scan

A scan focused on mapping firewall rules rather than port states directly.

`sudo nmap -sA -p- 192.168.1.1` (repeat for others).

Use this after a SYN scan. Which ports appear filtered? Discuss firewall implications.

Inverse TCP Flag Scan (TCP Null Scan)

A broad term for scans using invalid or no flags (inverse to normal TCP), often referring to Null scan (-sN), but encompassing FIN/XMAS too—here,

`sudo nmap -sN -p- 192.168.1.1` (repeat for others).

Compare to other inverse scans (XMAS, FIN). Why are these called "inverse"?

Outputs

`sudo nmap -oN scan_results.txt 192.168.1.1`

Outputs scan results to a normal (human-readable) file.

To help you Understand futher – Discuss with your Class:
Can you Identify the Scans below? What scans are done below?

```
(kali㉿kali)-[~]
$ nmap -sX -T4 10.10.10.16
You requested a scan type which requires root privileges.
QUITTING!

(kali㉿kali)-[~]
$ sudo su
[sudo] password for kali:
(root㉿kali)-[/home/kali]
# nmap -sX -T4 10.10.10.16
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-21 02:11 EST
Nmap scan report for 10.10.10.16
Host is up (0.00034s latency).
All 1000 scanned ports on 10.10.10.16 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 00:0C:29:1D:C0:EF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.70 seconds

(root㉿kali)-[/home/kali]
# nmap -sX -T4 10.10.10.16
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-21 02:12 EST
Nmap scan report for 10.10.10.16
Host is up (0.00049s latency).
All 1000 scanned ports on 10.10.10.16 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)
MAC Address: 00:0C:29:1D:C0:EF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 21.66 seconds
```

a.

```
(root㉿kali)-[/home/kali]
# nmap -sU -T5 10.10.10.16
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-21 02:21 EST
Warning: 10.10.10.16 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.10.16
Host is up (0.00046s latency).
Not shown: 988 open|filtered udp ports (no-response)
PORT      STATE SERVICE
19/udp    closed chargen
137/udp   open  netbios-ns
161/udp   open  snmp
3389/udp  closed ms-wbt-server
18543/udp closed unknown
19315/udp closed keyshadow
21902/udp closed unknown
26407/udp closed unknown
38498/udp closed unknown
41058/udp closed unknown
41896/udp closed unknown
49503/udp closed unknown
MAC Address: 00:0C:29:1D:C0:EF (VMware)
```

b.

```
(root㉿kali)-[/home/kali]
# nmap -Pn -p 80 -sI 10.10.10.16 10.10.10.10
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-21 02:22 EST
Idle scan using zombie 10.10.10.16 (10.10.10.16:80); Class: Incremental
Even though your Zombie (10.10.10.16; 10.10.10.16) appears to be vulnerable to IP ID sequence prediction (class: Incremental), our attempts have failed. This generally means that either the Zombie uses a separate IP ID base for each host (like Solaris), or because you cannot spoof IP packets (perhaps your ISP has enabled egress filtering to prevent IP spoofing), or may be the target network recognizes the packet source as bogus and drops them
QUITTING!
```

c.

```
(root@kali)-[/home/kali]
# nmap -Pn -p 80 -sI 10.10.10.10 10.10.10.16
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-21 02:23 EST
Idle scan using zombie 10.10.10.10 (10.10.10.10:80); Class: Incremental
Nmap scan report for 10.10.10.16
Host is up (0.0068s latency).

PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 00:0C:29:1D:C0:EF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.08 seconds
```

d.

```
(root@kali)-[/home/kali]
# nmap -sP 10.10.10.*
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-21 02:23 EST
Nmap scan report for 10.10.10.1
Host is up (0.00088s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 10.10.10.2
Host is up (0.00034s latency).
MAC Address: 00:50:56:F4:76:12 (VMware)
Nmap scan report for 10.10.10.10
Host is up (0.00028s latency).
MAC Address: 00:0C:29:60:3C:DA (VMware)
Nmap scan report for 10.10.10.16
Host is up (0.00034s latency).
MAC Address: 00:0C:29:1D:C0:EF (VMware)
Nmap scan report for 10.10.10.254
Host is up (0.00044s latency).
MAC Address: 00:50:56:E9:75:26 (VMware)
Nmap scan report for 10.10.10.11
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 2.17 seconds
```

e.

Observe the output. What scans are these?

1. ?
2. ?
3. ?
4. ?
5. ?

Further Study Pathway:

- + **Assess Vulnerabilities**
- + **Enumerate the Network**
- + **Exploit the Network**
- + **Mitigate the Vulnerabilities**
- + **Harden/ Secure the Network**