CSEC.140.603: Term Paper

Network and Service Security

13.04.2025

Mehrin Fathima, 418006818

Muhammad Ammar Rashid, 421007820

Reema Naheem, 397000990

Sameera Jasmine, 421008418

Department of Computing Security, Rochester Institute of Technology Dubai

TABLE OF CONTENTS

ABSTRACT

Digital network communication depends on network and service protection as its core cybersecurity element to maintain protection and reliability for inter-device and service operations. The security system applies multiple security layers to stop unauthorized entry and data breaches as well as service interruption. The fundamental aspect of network security exists in data-in-transit protection through encryption protocols and secure communication protocols that deliver confidentiality and integrity. The combination of firewalls with intrusion detection systems permits traffic oversight and control to block doubtful behaviour and defend against unauthorized entry. The protection of network-based applications and services requires service security to manage vulnerability exposure and access control definition and authentication protocols. Protection measures for network ports require regular software maintenance as well as disabling services that are no longer needed or maintain inadequate security. The adoption of protocols that use SSH and HTTPS has replaced dangerous outdated protocols such as Telnet and FTP which provides encryption for communication. Network and service security join forces to create trust boundaries and sustain system availability along with following security standards. The system needs continuous security updates to protect against new types of threats that emerge. Network and service security stands essential for digital organizations because it provides protective measures to maintain digital resilience and protects assets and enables secure connections across different user platforms [1]. In this paper, the most significant aspects of network and service security like protocols, attack vectors, and mitigation techniques will be covered. It will explore network traffic, port defense, old protocols, and network services like DHCP and DNS. Additionally, the paper includes a hands-on DNS spoofing attack using ARP cache poisoning to simulate a man-in-the-middle attack, explain its impact, and discuss mitigation strategies.

KEYWORDS*:* Network security, Service protection, Encryption, Firewalls, Secure protocols, Access control

INTRODUCTION

NETWORK AND SERVICE SECURITY

The Network Security Services collection offers a collection of cryptographic computer libraries that facilitate the development of security-enabled applications on many platforms. Additionally, the collection offers optional hardware-based TLS/SSL server speed enhancements and client smart card verification. A methodological defence system against online threats requires network and service security to function as fundamental building components. The domain establishes required security systems to defend device ports together with network traffic. Reliable safeguard systems which guard sensitive data depend on the identification of protective measures.[2]

Network Traffic: Composition and the Need for Protection

Data flow across a computer network is known as network traffic, including all means of communication between any device, server, or services. The data is sent in small units called packets and passed throughout the network then reassembled at the receiving system. We live digital lives, digitizing our daily activities such as web browsing, downloading files, PDFs, emails, voice and video calls, using cloud applications etc., which results in network traffic. The process of these activities leads to packets containing a vast array of contents, such as sensitive personal information (e.g., login credentials, financial information) as well as business data (e.g., confidential documents, company secrets) and control messages for connected systems and services.

Cyberattacks often focus on network traffic as network traffic normally contains sensitive and confidential information. Because packet sniffing tools can be used to intercept unsecured traffic, attackers can capture private data. Attacks such as man-in-the-middle (MitM) one can go further and can alter or inject malicious content into the data bitstream. Traffic levels above (or below) average and the patterns of that traffic can be indicative of threats such as Distributed Denial of Service (DDoS) attacks. Network traffic protection is thus therefore required to maintain data confidentiality, which means that data may not be seen by the users that should not see them; and integrity, which means that data are undisturbed in transit; and availability which means that the resources of the network are not interrupted.

These actions are all taken to safeguard the network traffic that they are a part of; thus, organizations deploy security measures such as firewalls, encryption protocols (like HTTPS or VPNs), and intrusion detection systems (IDS). These tools ensure that no unauthorized access can take place and detect suspicious activity and thus protect data against exposure or compromise. It also helps to monitor, and analyse, the patterns of traffic to detect, and manage, against threats early. To sum up, cybersecurity has evolved to a point where protecting network traffic is imperative and is crucial for communication that is secure, private, and free of any network integrity.[3]

Ports on Devices and Their Role in Security Vulnerabilities

WHAT ARE PORTS ON A DEVICE?

Ports are a virtual communication channel in computer networking that serves as endpoints for transferring data over a network from among the devices, apps, and services. Some of numbered ports run from 0 to 65535 on each computer or server which is linked to the internet. They speak to identify apps or a network service via these port numbers. Ports 80 and 443 are examples of ports used to transfer HTTP and HTTPS web traffic respectively. Additional commonly used ports are port 3389 for RDP, port 21 for FTP, port 22 for SSH, and port 25 for SMTP (email).

The port can be either open, closed or filtered, depending on if the service is active in that port or device firewall is configured. Services must be accessible so they can be open ports, but open ports are also potential attack surfaces. Attackers can execute, or try to execute, unauthorized access or destroy operations if a vulnerable service is running on an open port. For example, open SMB ports (445) have been used in hacker scale attacks like WannaCry, and Telnet (port 23) is a high-risk service as it offers no encryption.

While these ports are necessary for normal operations, they must be closely supervised and monitored. Reduction of the number of threats can be ensured in the use of unused ports, applying of firewall rules and use of secure protocols. Good system management at the port level is not just important as it is a guarantee that the system switches are working properly; it is also very critical to counter any intrusion, malware infection and leak of private information.

EXPLOITATION OF PORTS BY HACKERS

Ports are essential to network functionality but can also be a vulnerability compared to uncontrolled open or insecure ports. Port scanning tools are often used by hackers to try and determine if an open port is present on a target system. So, after being discovered, they may exploit known vulnerabilities related to the services that run on these ports. For example:

- **Port 21 (FTP):** The FTP (Port 21) is an unencrypted protocol, which can be attacked with brute force, rotten drawing, man in the middle.
- **Port 22 (SSH):** Although SSH is encrypted, if not appropriately secured, SSH can be exploited by attackers who have leaked SSH key or have pawned a credential through brute force attacks.
- **Port 23 (Telnet):** Telnet transmits data in plaintext, hence vulnerable to credential sniffing and spoofing attacks.
- **Port 445 (SMB):** SMB: Port 445 has been utilized in some high-profile attacks such as WannaCry, where vulnerabilities in the SMB protocol for example, are exploited to spread ransomware.
- **Port 3389 (RDP):** RDP ports are an attack surface points for credential stuffing attacks that may result in unauthorized access and ransomware deployment.

Understanding models and how to restrict open ports that are accessible to prevent such breaches or potential access from unauthorized individuals is very important.[4]

Strategies and Techniques to Protect Network Ports

One of the key parts of network protections is to stop users from accessing network and device services that are not intended for them, as this is done to prevent users from performing any cyberattacks or methods to access any devices or services. Ports being the place where network communication comes in, must be guarded carefully to reduce the level of threat posed.

Closing all unnecessary or unused ports is one of the most useful strategies. The problem with leaving ports open by default, especially in cases where the ports do not serve a known purpose, is that it makes the doors open for potential attackers to find a weakness. Never leave open ports that have no intention of use because this increases the attack surface and has no operational value. Port scanning is the first step for many cyberattacks as the attacker tries to determine which ports are open and then attempts to probe for vulnerable services that reside on them.

Maintaining strong cybersecurity depends on putting sensible plans into action to guard these ports.

1. **Close Unused Ports:** You need to close all ports except the port which is active. Open ports are unnecessary because they increase attack surface, which gives hackers chances to infiltrate. It is by default that you should only leave open only those ports that have to be open for some service and not all of them.
2. **Regular Port Scanning and Monitoring:** Perform periodic scans for open ports and determine how much of their usefulness are they needed. Detected unexpected open ports using tools like Nmap can be remediated on time.
3. **Implement Firewalls and Access Control Lists (ACLs):** While we do not talk too much about it here, one practical way to implement Firewalls and Access Control Lists (ACLs), is through Docker containers. ACLs are used to identify systems or users that can access given network resources and restrictions can be set based on which ports traffic is allowed. The approach is layered to only allow communication through designated ports by authorized entities.
4. **Use Secure Protocols:** Use secure alternatives to the outdated and insecure protocols such as SSH and SFTP to replace outdated and insecure protocols like Telnet and FTP. In secure protocol one encrypts data in transit, this means the data cannot be watched for and do not belong to anyone.
5. **Regularly Update and Patch Systems:** Always make sure you have your latest system and service security patches up to date. If services listening on open ports are failed to be remediated quickly, this can be exploited.
6. **Employ Intrusion Detection and Prevention Systems (IDPS):** Network traffic monitored by IDPS tools for suspicious activities impact the network traffic by providing real time alerts and responds automatically to potential threats which can be accessed through open ports.
7. **Utilize Port Knocking and Single Packet Authorization:** These techniques close the ports until certain streams of network packets are received, thereby concealing services to unwanted users.

Should Unused Ports Be Open by Default?

Ports, by default, should not be open if it does not have an intended use. Port scanning techniques reveal open ports, and associated services can be vulnerable or may be

misconfigured and be exploited. Closing of unused ports reduces the number of attack vectors for the overall network security. Beyond Trust points out that port usage understanding and management theft are critical elements of a solid security position.

Deprecated and Vulnerable Protocols

Several older protocols used today are deemed insecure and deprecated as they are simply weakened by inherent vulnerabilities with these weaknesses including lack of encryption, weak authentication, and susceptibility to modern attack techniques. Below are three major protocols that should no longer be in use, their respective ports, and a little about why they are vulnerable.

Telnet (Port 23)

- **Overview:** Telnet got its beginning in the 1960s and is one of the first protocols on which remote terminal access was built. It enables user connection to a remote device and issuing text-based commands to it.
- **Why It's Vulnerable**: Telnet is not encrypted so usernames, passwords, …, all data sent out on a Telnet session is all plain text. That's why a packet sniffing tools (Wireshark) attacker can easily intercept and read the sensitive information very easily. Furthermore, it has not built in authentication verification giving attackers the ability to spoof sessions or hijack a legitimate connection.
- **Consequences of Use**: Telnet used on the public or enterprise networks to enter for commands can allow unauthorized access, credential theft, and the manipulation of the system. Using it over the untrusted networks such as the internet is very dangerous in particular.
- **Secure Alternative**: However, because SSH (Secure Shell) encrypts all communication and includes strong authentication features it is a much safer method of remote access.

FTP (File Transfer Protocol) – Port 21

- **Overview**: Transfer of files between client and server systems over a TCP network is done by FTP. This protocol has also been widely used for decades in web hosting and enterprise systems.

- **Why It's Vulnerable**: Like Telnet FTP transmits data in plain text, so files contents, usernames and passwords are all visible. Moreover, it provides no data integrity check, and no strong authentication mechanisms, thus susceptible to:
  - Sniffing attacks
  - Man-in-the-middle (MitM) attacks
  - Brute-force login attempts
  - The use of open FTP servers to port scan internal networks through FTP bounce attacks
- **Consequences of Use:** Sensitive documents can be intercepted, credentials harvested, or open FTP servers on the network can be used as pivot points into more secure areas of the network.
- **Secure Alternative**: Encryption, authentication and better security controls are provided by SFTP (SSH File Transfer Protocol) or FTPS (FTP over SSL/TLS).

SNMPv1 and SNMPv2 (Ports 161/162)

- **Overview:** The monitoring and controlling network devices like routers, switches and servers are done using SNMP (Simple Network Management Protocol).
- **Why They're Vulnerable:** Currently, community strings for access are used in SNMPv1 and v2 and are traditionally left at the default values like 'public' or 'private'. Of course, encryption isn't part of the picture, so device info, configuration data, traffic can be intercepted. Also, attackers can change networks configuration, or extract information such as the routing table, and user accounts.
- **Consequences of Use:** Device compromise and network mapping are possible if snooping on SNMPv1/2 is performed, and Denial-of-Service (DoS) becomes possible if sensitive parameters are changed or overloaded.
- **Secure Alternative:** Strong authentication and encryption are introduced in SNMPv3 to provide both data in transit and access credential security.

PROTOCOL	PORT	MAJOR VULNERBALITIES	RECOMMENDED REPLACEMENT
Telnet	23	No encryption, plaintext data, spoofing	SSH
FTP	21	Data sniffing, bounce attacks, plaintext login	SFTP, FTPS
SNMPv1/2	161/162	Default credentials, no encryption, easy interception	SNMPv3

Old protocols phasing out and replacing it with newer, more efficient and safer protocols lowers the level of puts an organization's data at risk drastically, and controls data protection in the network.[5]

RELATED WORKS

Article 1*: DHCP attacking tools: an analysis*

Dynamic Host Configuration Protocol (DHCP) is a core networking protocol that was developed to dynamically assign IP addresses and other related network configuration, such as subnet mark and default gateway to devices on a network, using a four-step process called DORA: Discover, Offer, Request, Acknowledge. It operates over UDP using port 67 for the server and 68 for the client. DHCP is considered a vulnerable and insecure service as the protocol does not require authentication from the DHCP client. The paper conducts a thorough study of publicly known DHCP attacking tools, that are commonly used to target the DHCP service, to aid in the detection and mitigation of such attacks. DHCP services can be attacked in various ways, including DHCP flood attacks, DHCP starvation attacks, and DHCP spoofing attacks. One of the most common attacks is DHCP flooding attack, a form of Denial-of-Service (DoS), where the attacker repeatedly sends forged DHCP requests to disrupt the protocol's functionality and performance. During the attack, a massive number of DHCPDISCOVER packets are sent to the server without completing the DORA cycle causing the server to reserve many IPs, which are never leased, disrupting DHCP responsiveness and availability. The tools that are commonly used to carry out this attack include DHCPwn, Yersinia and Hyenae. Some of the mitigation strategies discussed in this article include DHCP snooping with Rate Limiting, in which abnormal DHCP traffic is detected, dropping excessive requests, and Signature-Based Detection, where the attack patterns are identified utilizing characteristics like repeated XIDs, MAC mismatches, or bogus hostnames. Another form of DoS is the DHCP starvation attack, a special kind of flood attack, where an attacker continuously sends forged DHCP client messages to target the server's pool, aiming to exhaust the server's available IP addresses. The fake requests sent by attacker complete the full DORA cycle, tricking the server to lease all available IP addresses to non-existent clients. As a result, legitimate clients are unable to obtain IP addresses, leading to a loss of network connectivity. The article analyses tools that are commonly used for this attack, which includes DHCPig, dhcpstarv, Dstar, and Hyenae, all of which use the full DORA cycle to lease IPs. Common mitigation strategies used against DHCP starvation is DHCP snooping, in which only trusted ports are allowed to send DHCP responses and MAC addresses that mismatch are blocked, and Port Security, which limits the number of MAC addresses allowed per switch to prevent mass fake requests. In conclusion, the article provides a detailed examination of widely known DHCP attacking tools, laying the groundwork for developing stronger, more effective countermeasures against DHCP attacks. For future work, the authors plan to implement and test a comprehensive detection and mitigation system based on their findings [11].

Article 2: *DNS Advanced Attacks and Analysis*

DNS or Domain Name System is a critical networking protocol which is primarily responsible for mapping domain names to corresponding information, like IP addresses. It allows users to find resources on network by converting human-readable domain names to machine-readable IP addresses. DNS also plays a key role in service discovery, load balancing, and email routing. The article conducts a detailed study on the growing importance of DNS in modern network systems, the possible attacks on DNS servers and their potentially devastating impacts, and finally the mitigation strategies that can be implemented to protect DNS against such cyber-attacks. DNS systems are vulnerable to several attacks, including DNS cache poisoning attack, Amplification attacks, DNS hijacking and DoS attacks. While the article analyses all these attacks in detail, this paper focuses specifically on two spoofing-based attacks—DNS cache poisoning and DNS hijacking, as they are most relevant to the concept of DNS spoofing and form the basis of the analysis and demonstration described in the following sections. DNS cache poisoning occurs when the attacker injects malicious DNS data into the cache of a DNS resolver. When the client requests a domain name, the poisoned DNS resolvers return forged IP addresses, often leading the user to fake or malicious websites. Once the attacker redirects the incoming traffic to a server of their choice, they will be able to launch additional attacks or collect confidential information from the victim. Some of the mitigation strategies against cache poisoning include DNSSEC (Domain Name System Security Extensions), which adds digital signatures to DNS data, ensuring authenticity of websites, and limiting recursion to internal users to limit exposure to external attacker. Another common attack on DNS systems is DNS hijacking, where the attacker introduces a rogue DNS server under their control or modify the functioning of a real DNS server, which is then used for malicious purposes, like phishing. During the attack, incoming traffic is intercepted or redirected (Man-in-the-middle attack), leading to lose of trust and DNS reliability. DNS hijacking can also affect secure browsing and VPNs. To prevent or mitigate these attacks, DHCP snooping and trusted DNS settings could be used, which ensures that devices receive DNS settings only from trusted DHCP servers. Although the article analyses DNS attacks and their mitigations in a deeper level, the authors aim to investigate more security challenges that come with advanced DNS attacks [12].

Article 3*: A Survey of Email Service; Attacks, Security Methods and Protocols*

Email is an essential tool in today's digital world, which is used for a wide range of purposes, from simple personal communication to more formal and confidential communication in workplace and business environments. SMTP (Simple Mail Transfer Protocol) is a standard network protocol that is used for sending and forwarding emails over IP networks. Although SMTP is used for global communication, it was not originally designed with strong security features, making it vulnerable to various cyber-attacks. The article provides an in-depth study of common threats in email services and their potential mitigation strategies. Two common attacks that target SMTP protocol during email communication are email spoofing and phishing attack via SMTP. Email spoofing occurs when the attacker forges the "from" address to make it appear as though it originated from an authentic source, even though it was sent from somewhere else. This form of spoofing is easy to conduct as SMTP lacks in-built authentication feature. Mitigation strategies discussed in the article include Transport Layer Security and Secure Sockets Layer (TLS/ SSL) which can be applied to ensure authentication. Phishing attacks can be conducted by exploiting SMTP to send phishing emails, deceiving users into revealing sensitive and confidential data like passwords, credit card details, etc. During the attack, users will be tricked into opening any attachments or links, that ask users to enter confidential information, which is used later for malicious purposes and can lead to dangerous consequences like identity theft or financial loss. The mitigation strategies include spam filters, that can decrease the number of incoming phishing emails, and educating users on how to identify phishing emails, which can significantly reduce the rate of success for phishing [13].

Article 4: *Multiple Vulnerabilities in SNMP*

The article discusses about SNMP (Simple Network Management Protocol), an application-layer protocol that is used for managing and monitoring network devices like routers, switches and servers, and the security issues associated with this protocol. It explains how the simplicity and outdates security mechanisms of SNMP and particularly versions of SNMP like SNMPv1, have made it vulnerable to several attacks. Since SNMP protocol is widely deployed, networks could be exploited with dangerous consequences. Among the common attacks on SNMP are malformed packet attacks and community string sniffing and spoofing. Malformed packet attacks occur when the attacker exploits the vulnerabilities in SNMP agents and NMS (Network

Management System) to send malformed or overly long SNMP messages with the intention of crashing the service or causing buffer overflows. This was exposed by the 'Protos test suite', which found issues in over 100 vendors' products. This type of attack is also known to cause DoS (Denial of Service) and unstable device behaviour. Some of the common mitigation steps that can be taken against these attacks include usage of IDS (Intrusion Detection System) with updated signatures to detect malformed packets and applying vendor patches and firmware updates to improve validation of SNMP messages. Another type of attack against SNMP is community string sniffing and spoofing, that exploit SNMP "community strings", a plaintext that is used by SNMP for authentication. Since community strings are easy to sniff on the network or guess, attackers read or modify the configurations. Additionally, UDP source addresses can be spoofed, allowing attackers to impersonate trusted systems. One way to mitigate this attack is to change default community strings to strong, unique values. SNMP scanners (e.g., SNMPing, SNScan) can also be used to identify vulnerable devices on the network. The article emphasizes the by implementing best practices such as updating software, changing default settings, and applying filtering rules, organizations can significantly reduce the risk of SNMP-based attacks [14].

Paper Reference:	***Tools Used:***	***Accuracy:***	***Results:***
[11]	The article analyses 7 known DHCP attacking tools and 2 packet crafting libraries. The tools analysed are DHCPig (Python), dhcpstarv (C), Dstar (C++), DHCPwn (Python), Yersinia (C), Hyenae (C), Ettercap (C) and Scapy & Kamene (packet crafting libraries).	The tools were tested in a controlled virtual environment and used Wireshark to analyse the traffic. The study compared legitimate vs attack traffic patterns and proposed a hybrid detection mechanism using fingerprint-based detection and behavioural signatures. The accuracy of the study was verified through signature matching and timing thresholds.	The attacking tools were classified by type, severity and attack patterns. The attack speed and bandwidth were measured and identified unique signature per tool. Yersinia showed the highest packet rate, while dhcpstarv was most persistent. The study proposes thresholds for detecting DHCP floods and starvation attacks and the recommended countermeasures against DHCP attacks are: DHCP Snooping, Port Security,

			authentication mechanisms.
[12]	No tools were tested in this paper, but it analyses botnets, open resolvers, and DDoS tools (e.g., Stacheldraht) as attack vectors. The paper also mentions DNSSEC, DNS Stealth Architecture, and SIEM tools (e.g., Savanture, UltraTools.com) as defense tools against DNS attack.	The paper mentions real world attack patterns and references security experts and industry practices. The accuracy of the article is supported by the logical analysis and citation of vulnerabilities and attack incidents.	Multiple DNS attack types were identified, like DNS Cache Poisoning, DNS Hijacking, Amplification Attack, and DoS/DDoS. The paper proposes detailed preventive measures which include DNSSEC, ACLs, stealth architecture, DNS monitoring, and outsourcing DNS. The best practices discussed in the paper are domain locking, access control, monitoring, and TTL adjustments.
[13]	The paper is survey-based and analyses email threats, security protocols, and filtering techniques. No tools were tested in the study but the author analyses some of the well-known tools and technologies like S/MIME, PGP , Bayesian Filters, Genetic Algorithms, Support Vector Machines (SVM), Fingerprint Authentication, Elliptic Curve Cryptography (ECC) and Email Filters.	The paper was not experimental and hence does not include experimental accuracy metrics. It theoretically compares the techniques and also references studies that reported high precision, like fraud detection model with 96% accuracy and spam filtering with over 82% effectiveness.	Several common email threats were identified, including phishing, spoofing, fraud, eavesdropping, and malware. The paper proposes several mitigation strategies and suggests that combining different solution methods (e.g., filtering + cryptography) can enhance safety and will be more effective in the long run.

[14]	The Protos SNMPv1 test suite was the main testing tool, that was used to discover vulnerabilities in SNMP implementations. Other tools mentioned in the article include SNMPing, SNScan, Snort IDS, Cisco Secure IDS, and ISS RealSecure for detection and scanning.	Protos suite detected flaws in over 100 products of vendors, verifying threats like buffer overflow and trap handling flaws. The study relied on systematic testing rather than statistical accuracy but was supported by strong industry backing via CERT.	The paper discusses serious SNMP weaknesses in authentication, input handling, and default configurations. It recommended mitigation strategies such as ingress/egress filtering, changing default community strings, applying vendor patches, and using updated IDS signatures to reduce exploitation risks.
Our Paper	In our paper, for the practical section, the tools used are Kali Linux, Ettercap, and Python for running a spoofed website. Ettercap was used to perform ARP cache poisoning and DNS spoofing, while the attacker's fake site was hosted using a Python web server. Additional tools like ifconfig, arp, and curl were also used for network manipulation and setup.	Accuracy was maintained by cross-referencing findings with existing literature and security advisories (e.g., CERT, OUSPG). Research answers were supported with examples and real-world attack methods. While most of the paper is theoretical, the later hands-on DNS spoofing test demonstrates practical accuracy by showing real-time effects of a MITM attack.	Our paper presents a comprehensive analysis of network and service security by tackling a number of attack surfaces (DNS, DHCP, SNMP, SMTP, and NTP). What distinguishes this paper is the inclusion of both academic comparison and simulation of a real-world attack, unlike most papers that are either theoretical or narrowly focused. This combination allows readers not only to learn but also observe how attacks like DNS spoofing can be carried out in test environments.

Table 1: Summary of all the related works summarized above and our paper.

NTP ATTACKS AND THEIR IMPACT ON FORENSIC ANALYSIS

NTP or Network Time Protocol is a network protocol that is responsible for maintaining consistent time synchronization across devices in a network. Accurate timekeeping is important for maintaining consistent log entries, coordinating events, and ensuring the integrity of forensic investigations. Attackers can target the vulnerabilities in the NTP servers to alter the time responses sent to devices on the network, causing the devices to accept wrong timestamps. This type of attack is called ***NTP spoofing***.

Studies have shown that NTP spoofing can have serious consequences by shifting system time drastically, even by years in some attacks (e.g. changing date to year 2040). This will create a significant disruption in forensic timelines- events are logged out of order, or appear to happen in the future or past, making it nearly impossible to reconstruct the attack sequence accurately. When multiple systems are involved, correlating logs across devices is unreliable, and important evidence may be disregarded due to invalid timestamps. This can adversely affect organizations that try to trace attacker's path through network, detect lateral movement, or understand when key systems were accessed. In more extreme cases, inaccurate timestamps can render logs inadmissible as legal evidence. Hence, it is extremely important to prevent and/or mitigate these attacks on NTP servers. To mitigate NTP spoofing, the study proposed a detection system that combines GNSS sensors (which use satellite time) with a Long Short-Term Memory (LSTM) machine learning model. This system learns what "normal" time synchronization looks like and can detect suspicious shifts, helping administrators catch and stop NTP spoofing attempts before they impact forensic analysis. Compared to other traditional mitigation methods, like firewall-based filtering or network packet authentication, the proposed solution responds better to timing changes and can also offer an extra strong layer of security [15]

PROPOSED SCHEME

In the following demonstration, Kali Linux was used to simulate a DNS spoofing attack in a controlled environment. The activity showcases the importance of DNS spoofing integrity and the vulnerabilities that can arise when DNS traffic is not properly secured. By various stages such as network configuration, ARP cache poisoning and spoofed DNS responses etc. This illustrates how attacker can manipulate DNS requests to redirect a victim to a malicious site. This activity has key concepts such as man-in-the-middle-attacks, packet forwarding and how the exploitation of unsecured local networks is explored and understood in practice.

Updating Kali Linux Packages

Before starting the attack, the Kali Linux device was updated to ensure that all packages and dependencies were current (Figure 1).

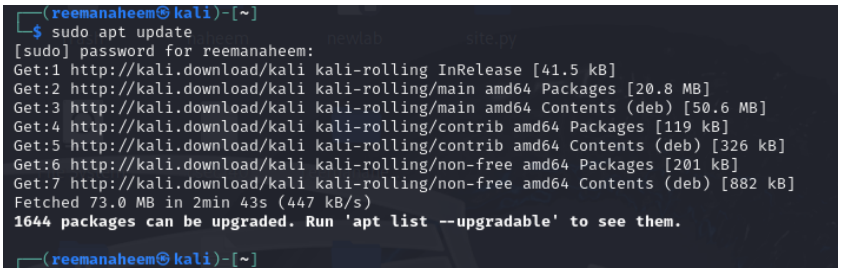

Figure 1: Using sudo apt update to update Kali Linux

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

During the spoofing, the victim would later see this website.

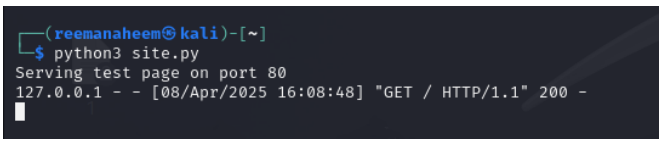

Figure 2&3: Using port 80 to serve the malicious website locally.

A browser was used to open 127.0.0.1 to confirm that the website was operational (Figure 3).

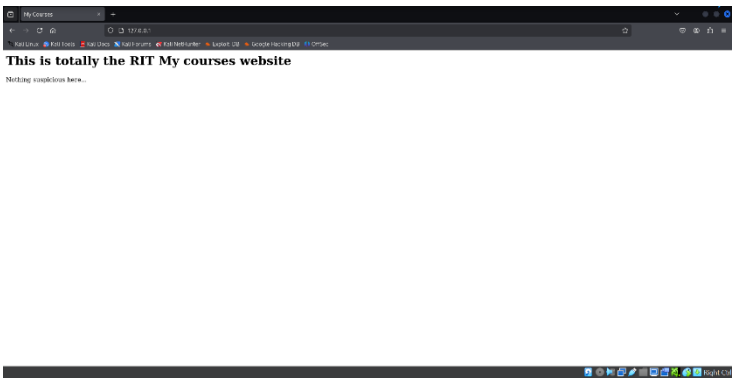

Figure 3: The spoof website launched locally in the attacker's browser.

Editing Ettercap's DNS Configuration File

In Ettercap's DNS configuration file, the domain to be spoof (in our test, mycourses.rit.edu) was routed to the attacker's IP after the attacker's IP address was verified using ip a.

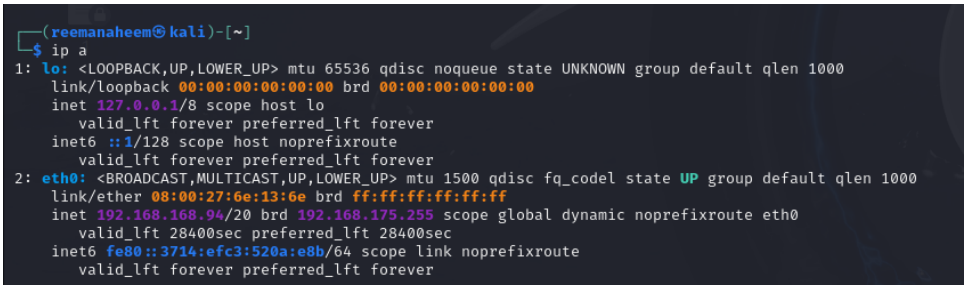

Figure 4: Ettercap DNS entry editing.

IP forwarding was enabled in Figure 5&6 to enable packet forwarding by Kali between network interfaces. Activating Kali's IP forwarding.

Figure 5&6: Enabling IP forwarding in Kali.

The main purpose of enabling IP forwarding in Kali is to permit devices to route traffic between various networks. This facilitates communication between networks by allowing the device acting as a router to forward packets between them. This shows that communication between networks would not be made possible without enabling IP forwarding. [6] While this function is essential for network routing, it is simultaneously also hazardous, due to the possibility of exploitation in a Man-in-the-Middle attack if the network connection is insecure. [7] If a Man-in-the-Middle attack were to occur, it would allow the attacker (an outer party) to alter and manipulate the routed traffic, which would lead to the packets of data being transmitted to the attacker first before reaching the other computers. [6][7]

Viewing and clearing the ARP Cache on the Victim Machine

The MAC and IP addresses of the router were resolved by Figure 7 ARP Cache in Ubuntu on the Ubuntu virtual machine. 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.

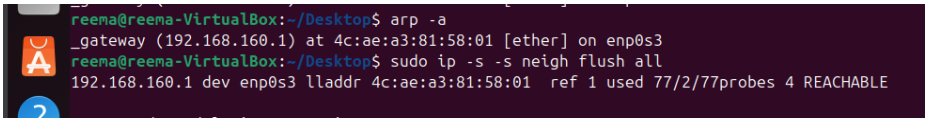

Figure 7: ARP Cache in Ubuntu.

Launching Ettercap for ARP and DNS Spoofing

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

Figure 9: Running Ettercap in terminal for ARP and DNS spoofing.

Commands:

In this step of ARP and DNS spoofing, Ettercap is the only command used. Ettercap is a network security tool used to protect against Man-in-the-Middle attacks on a Local Area

Network. It supports sniffing live connections, content filtering, and network protocol analysis. [6][7]

Options and Arguments:

- **-T**: This option is used to permit Ettercap to run in text-only interface mode. [7]
- **-M arp:remote**: This option is used by hackers when commanding Ettercap to perform a Man-in-the-Middle attack using ARP poisoning across remote hosts. **-M** is the option used to place Ettercap between two different hosts, facilitating the interception and alteration of the traffic flowing between the hosts. On the other hand, **-arp:remote** is the argument that enables poisoning even if the target hosts are not in the same subnet. [7]
- **-P dns_spoof**: This is used to load the **dns_spoof** plugin, which intercepts DNS requests and can redirect them to fake IP addresses set in the etter.dns configuration file. Here, **-P** is the option that tells Ettercap to use a plugin, while **dns_spoof** is the argument that specifies which plugin is to be used, that being the DNS spoofing plugin. [7][8]
- **-i eth0**: Here, **-i** is the option used to specify the network interface intended for use, whereas **eth0** is the argument that clarifies which network interface the attacker's machine is using. Ettercap makes use of **–i eth0** to sniff and inject packets when intercepting traffic. [6][7]
- **/VICTIM_IP//:** This is the argument that tells Ettercap to target the victim. Attackers enter the intended victim's IP address to perform the attack. [7]
- **/ROUTER_IP//:** This argument tells Ettercap to target the Local Area Network (LAN)'s router. Similar to **/VICTIM_IP//**, attackers enter the target router's IP address to perform the attack. [7]

In both **/VICTIM_IP//** and **/ROUTER_IP//**, the double slashes are used to separate the IP address of multiple targets within the network. [7]

This command sets up a Man-in-the-Middle attack scenario, because Ettercap has successfully poisoned the victim and the router's ARP caches. Now, both the victim and the router would falsely believe that the attacker's MAC address belongs to the other. Poisoning both caches is essential for successfully intercepting and manipulating traffic. Otherwise, the traffic would simply flow from the router to the victim without reaching the attacker. [6][7]

Verifying ARP Table Spoofing on the Victim Machine

To confirm that ARP spoofing had been successful, the victim system pinged the router's IP address to access ARP communication. As shown by Figure 9, the command arp -a was executed to examine ARP cache. The MAC address of the router's IP address had been spoofed and now showed the MAC address of the Kali system, proving that the victim had been spoofed to forward traffic to the attacker with complete success.

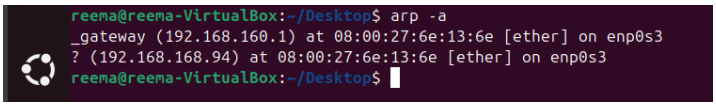

Figure 10: Altered ARP Table on Victim Machine After ARP Spoofing

As observed in the previous command, the attacker used Ettercap with ARP spoofing, which led to the victim and the router falsely believing the attacker’s MAC address belongs to the other device in their network. [7] Due to this, when the victim opens their ARP table, the table is updated to associate the router’s IP address with the attacker’s MAC address. Therefore, as a result, all the network traffic goes through the attacker’s machine before reaching the actual router. This shows that the attacker has successfully performed a Man-in-the-Middle attack. [6][7]

Victim Accesses the Spoofed Domain

Meanwhile, the Python script continued to run the fake website, ensuring that any spoofed requests would land on the malicious site.

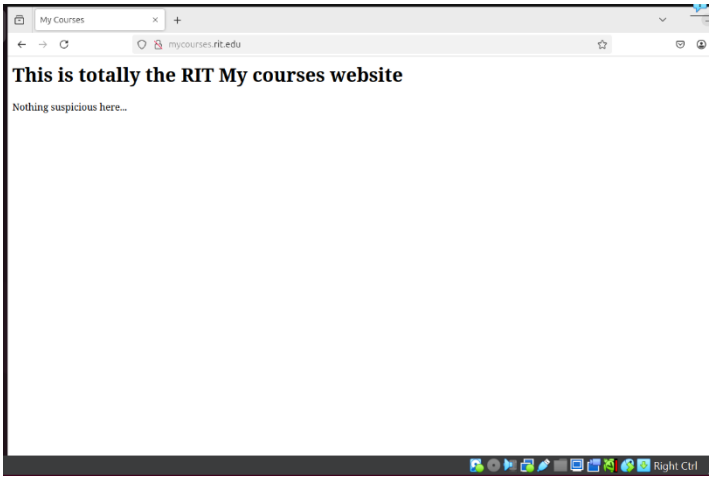

Figure 11: Victim redirected to the spoofed website.

Once the victim accessed the targeted domain (e.g., mycourses.rit.edu), the spoofed website appeared instead of the legitimate one, demonstrating a successful DNS spoofing attack.

The type of cyberattack presented in the above scenario is known as phishing. Since the URL is legitimate and the layout of the website is convincing, the victim could be fooled into believing that this is the website they intended to access. Therefore, due to being redirected to this website, the victim has unintentionally shared sensitive information with the attacker, such as the username, password and personal details. The attacker could use this information to perform identity theft and in other cases, take part in online transactions using the victim's credentials without the user's knowledge. In addition, such an attack can steal other forms of personal data as well, discreetly observe the victim's activity, damage the overall system, etc. [6]

The most serious types of cyberattacks involve the following scenarios among others:

- **Financial theft:** Users are redirected to fake banking sites through which they share their credit card details, putting them at risk of unauthorized transactions or stolen funds. [9]
- **Identity theft:** Fake login pages can be accessed by users, who then share their usernames and passwords, allowing the attacker to hijack their accounts. [9]
- **Malware Infection:** Fake advertisements, such as software updates, may lead to the user unknowingly downloading malicious software, which would then harm the computer. [7]

To prevent such attacks from happening, individuals can make use of a PIN or Multi-Factor Authentication, such as biometric scans, for example, as those details are private only to the user and therefore in this way, a potential attacker will not be able to access the user's personal information (such as banking details) or perform an attack. [6] People can also make use of DNS Security Extensions to protect their personal information. DNS Security Extensions are beneficial such that they assign cryptographic signatures to DNS data, which enables recipients of such data to verify its authenticity and integrity and prevents attackers from adding false DNS data into the user's cache. [8] Moreover, users can verify the integrity and authenticity of a certain website that they wish to access by checking if the protocol HTTPS (Hypertext Transfer Protocol Secure) is present in the website's URL, because this extension ensures that the website is secure and not vulnerable to attacks. [8]

CONCLUSION

This study emphasizes how exploitation of weaknesses in network protocols, like ARP, can be used by malicious actors to hijack traffic and mislead users, along with the real consequences of DNS spoofing attacks. By following a step-by-step implementation of the spoofing attack using Ettercap on Kali Linux, we demonstrated the ways in which attackers can display spoofed websites, alter DNS replies, and secretly tap into communications without being noticed instantly.

The illustration presented clearly depicts the risks of unsecured local networks alongside highlighting the key importance of using proper network security protocols, which include ARP inspection, DNS over HTTPS, and user education. Like Public Key Infrastructure (PKI), the DNS and ARP protocols have substantial attractiveness for hackers owing to their initial inability to provide strong security architectures.

In conclusion, creating effective protection requires comprehending and modelling these attacks. To guarantee the reliability and integrity of network communication, preventive defense mechanisms, enhanced encryption standards, and ongoing monitoring must be put in place as threats evolve.

REFERENCES

[1] *What is network security? definition and types*. Fortinet. (n.d.). https://www.fortinet.com/resources/cyberglossary/what-is-network-security#:~:text=Network%20security%20refers%20to%20the,edge%20and%20inside%20the%20perimeter.
[2] Wikipedia contributors. (2025, April 4). *Network Security services*. Wikipedia. https://en.wikipedia.org/wiki/Network_Security_Services#:~:text=Network%20Security%20Services%20(NSS)%20is,cards%20on%20the%20client%20side.
[3] Barney, N., & Lutkevich, B. (2022, October 5). *network security*. Search Networking. https://www.techtarget.com/searchnetworking/definition/network-security
[4] *What is network traffic? definition and how to monitor it | Fortinet*. (n.d.). Fortinet. https://www.fortinet.com/resources/cyberglossary/network-traffic#:~:text=Network%20traffic%20is%20the%20amount,the%20receiving%20device%20or%20computer.
[5] *Disabling unnecessary services and protocols | CERT NZ*. (n.d.). CERT NZ. https://www.cert.govt.nz/information-and-advice/guides/unused-services-and-protocols/disabling-unnecessary-services-and-protocols/
[6] *ARP spoofing | Imperva. (n.d.). Imperva.* https://www.imperva.com/learn/application-security/arp-spoofing/
[7] *Dewage, R. B. (2021, May 19). ARP spoofing and Ettercap usage. Medium.* https://ranmal-b-dewage.medium.com/arp-spoofing-and-ettercap-usage-bec8a2aebd15
[8] *DNS spoofing vs. DNS poisoning | ClouDNS. (2024, August 20). ClouDNS.* https://www.cloudns.net/blog/dns-spoofing-dns-poisoning/
[9] *Newman, L. H. (2017, April 4). Hackers hijacked a bank's entire online operation. WIRED.* https://www.wired.com/2017/04/hackers-hijacked-banks-entire-online-operation/
[10] *IP Routing Prevents Windows Client Authentication. (n.d.). Cato Networks.* https://support.catonetworks.com/hc/en-us/articles/13437113541149-IP-Routing-Prevents-Windows-Client-Authentication
[11] Aldaoud, M., Al-Abri, D., Al Maashri, A., & Kausar, F. (2021). DHCP attacking tools: An analysis. *Journal of Computer Virology and Hacking Techniques, 17*(2), 119-129. https://doi.org/10.1007/s11416-020-00374-8
[12] Hudaib, A. A. Z., & Hudaib, E. A. Z. (2014). DNS advanced attacks and analysis. *International Journal of Computer Science and Security (IJCSS), 8*(2), 63.
[13] A survey of email service; attacks, security methods and protocols. (2017). *International Journal of Computer Applications,* https://doi.org/10.5120/ijca2017913417
[14] Jiang, G. (2002). Multiple vulnerabilities in SNMP. *Computer (Long Beach, Calif.), 35*(4), supl2-supl4. https://doi.org/10.1109/MC.2002.1012421

[15] Romaniuc, A., Vasile, V., Borda, M., & Alexandru, B. (2024). NTP spoofing attack detection on time servers with GNSS sensors based on long short-term memory algorithm. Paper presented at the 1-6. https://doi.org/10.1109/COMM62355.2024.10741488