**Cyber security**

**Week 2**

**1. Deep Dive into Network Security**

 Task: Write a 12 page report explaining the roles of firewalls, VPNs, and IDS/IPS in securing a network. Use examples from realworld applications.

Network security is essential for organizations to protect sensitive data, communications, and infrastructure from cyberattacks. As networks become more complex and connected, the risks of unauthorized access, data theft, and disruption grow. To safeguard digital assets, key technologies like firewalls, Virtual Private Networks (VPNs), and Intrusion Detection/Prevention Systems (IDS/IPS) play vital roles. These technologies work together to provide layers of defense, ensuring that networks remain secure against both internal and external threats.

**Firewalls:**

Firewalls are network security devices or software designed to monitor and control incoming and outgoing network traffic. They create a barrier between an internal network and untrusted external networks like the internet. Firewalls filter traffic based on predefined security rules, allowing legitimate traffic while blocking malicious or unauthorized access.

**Types of firewalls:**

1. **Packet-Filtering Firewalls**: These firewalls inspect packets of data based on set rules but do not examine the entire packet. They are fast but provide limited security as they cannot detect complex attacks.
2. **Stateful Inspection Firewalls**: These firewalls maintain a record of active connections and make decisions based on the state of a packet and its context in a communication session, offering stronger security.
3. **Proxy Firewalls**: These act as intermediaries between end-users and the internet, providing more in-depth traffic inspection by processing requests on behalf of the user, protecting their identity and filtering malicious content.
4. **Next-Generation Firewalls (NGFW)**: Combining traditional firewall functions with advanced security features like application awareness and deep packet inspection, NGFWs can detect and respond to sophisticated threats like malware.

As a real world example of involving firewalls is the **2014 Sony Pictures hack**. While Sony had firewall defenses in place, the hackers used advanced techniques to bypass them. This event highlighted the importance of properly configuring firewalls and using next-gen solutions that integrate real-time threat intelligence.

Configuring a firewall involves setting rules to allow or block specific types of traffic. For instance, network administrators can configure firewalls to block traffic from certain IP addresses, only allow connections over specific ports, and enforce protocol restrictions. A common setup involves allowing HTTP/HTTPS traffic while blocking all others to protect web servers.

Firewalls can be compromised by misconfigurations or sophisticated attacks. Best practices include regularly updating firewall rules, enabling logging for traffic monitoring, and combining firewalls with other security measures (e.g., intrusion detection systems).

**Virtual Private Network (VPNs):**

A VPN creates a secure tunnel between a user and a network over the internet, encrypting the data transferred between them. VPNs are vital for organizations that need to protect sensitive information, especially in scenarios where employees work remotely or need to connect multiple office locations securely.

**Types of VPNs:**

1. **Remote Access VPN**: This type of VPN allows individual users to connect securely to a company's internal network as if they were physically present in the office. It’s particularly useful for remote workers.
2. **Site-to-Site VPN**: When different office locations need to securely connect over the internet, site-to-site VPNs are used to create a secure "tunnel" for data transfer between them.

**Encryptions Protocols Used in VPNs:**

1. **IPsec (Internet Protocol Security)**: IPsec encrypts and authenticates IP packets, making it a popular choice for site-to-site VPNs.
2. **L2TP (Layer 2 Tunneling Protocol)**: Often paired with IPsec, L2TP adds another layer of security for VPN connections.
3. **SSL/TLS**: These protocols are commonly used for securing web-based VPNs, encrypting traffic between a web browser and the server.

During the **COVID-19 pandemic**, companies like **Microsoft and Google** relied heavily on VPNs to secure remote work. VPNs ensured that employees could securely access corporate systems from home, without exposing sensitive data to cybercriminals.

VPNs can slow down network speeds due to the encryption process, and they are not immune to attacks if improperly configured. For example, using outdated encryption algorithms or weak passwords can compromise security.

**Intrusion Detection and Prevention Systems (IDS/IPS):**

IDS and IPS are security technologies that monitor network traffic for signs of malicious activity. IDS systems detect potential threats and generate alerts, while IPS systems take immediate action to block or neutralize threats. Together, they form a critical defense against sophisticated attacks like malware, unauthorized access, and insider threats.

**Types of IDS/IPS:**

1. **Network-Based IDS/IPS (NIDS/NIPS)**: These systems monitor entire network segments for suspicious traffic.
2. **Host-Based IDS/IPS (HIDS/HIPS)**: These systems monitor individual devices or hosts, providing security at the endpoint level.

IDS systems use known attack signatures and behavioral analysis to detect threats. If a potential attack is identified, the system generates an alert for network administrators. IPS systems take immediate action by blocking or redirecting suspicious traffic, effectively stopping threats in their tracks.

A good example of IDS/IPS in action is **Amazon Web Services (AWS)**, which uses these systems to protect its vast cloud infrastructure. With millions of customers and continuous data flow, AWS relies on IDS/IPS to identify and mitigate threats like Distributed Denial of Service (DDoS) attacks.

One challenge with IDS/IPS is the potential for false positives—alerts triggered by normal network activity. Fine-tuning the systems, updating threat signatures regularly, and using them in combination with other tools help minimize unnecessary alerts while maintaining robust security.

**Encryption Protocols in Network Security:**

Encryption is crucial for keeping sensitive data safe from prying eyes. It works by converting readable data into a coded format, which can only be deciphered by someone with the proper decryption key. Encryption protocols ensure that data remains confidential as it travels across public networks.

**Common Encryption Algorithms:**

1. **Symmetric Encryption (AES)**: AES is one of the most widely used encryption standards. It's fast and highly secure, making it ideal for protecting data transmitted over VPNs or stored on devices.
2. **Asymmetric Encryption (RSA)**: RSA is often used in VPNs and SSL/TLS protocols for secure key exchanges. Unlike symmetric encryption, RSA uses two keys—a public key to encrypt data and a private key to decrypt it.

Encryption is essential for securing VPN tunnels and protecting data exchanged between networks. Even if a hacker intercepts encrypted traffic, they won’t be able to read it without the decryption key. In the case of firewalls, encryption helps ensure that communication between trusted networks remains secure.

**Comparison:**

**Firewalls vs. IDS/IPS:**

Firewalls act as a gatekeeper for your network, controlling what data is allowed to enter or leave. They use a set of rules to block or allow traffic based on IP addresses, protocols, or ports. Essentially, they focus on keeping unwanted visitors out by filtering incoming and outgoing traffic.

On the other hand, IDS (Intrusion Detection Systems) and IPS (Intrusion Prevention Systems) work behind the scenes within the network. Their role is to watch for suspicious behavior that might have slipped past the firewall. IDS detects potential threats and raises alerts, while IPS goes one step further by not only detecting but also actively stopping those threats in real-time.

In simple terms: firewalls stand at the door to keep trouble out, while IDS/IPS are inside the house watching for anything strange happening once someone’s in.

**VPNs vs. Encryption:**

A VPN (Virtual Private Network) allows you to send and receive data securely over a public network, like the internet. It does this by creating a "tunnel" between your device and a remote server, hiding your real location and making your online activities private.

Encryption, on the other hand, is what makes the VPN tunnel safe. It scrambles the data being sent, turning it into an unreadable format so that even if someone manages to intercept it, they can’t make sense of it.

In a simple analogy, think of a VPN as the safe tunnel your data travels through, and encryption is the lock that ensures no one can open the box inside that tunnel. The two work together to protect your sensitive information from prying eyes. Without encryption, the VPN tunnel wouldn’t be secure.

**Future trends in Network Security:**

**Emerging Threats:**

As technology evolves, so do the tactics of cybercriminals. Traditional security tools, like basic firewalls and antivirus programs, are becoming less effective at defending against sophisticated attacks. Two major threats that are on the rise are **ransomware** and **Advanced Persistent Threats (APTs)**.

* **Ransomware** is a type of malware that locks or encrypts a victim’s files, demanding a ransom for their release. Attackers are now using more aggressive methods, such as targeting critical infrastructure, healthcare systems, and even governments, causing widespread disruptions.
* **APTs**, on the other hand, are more stealthy. These attacks involve hackers gaining unauthorized access to a network and remaining undetected for long periods, often extracting sensitive information over time. The goal isn't immediate destruction but long-term surveillance and data theft.

Because of these evolving threats, companies are being forced to rethink their security strategies. Static security measures, like rules-based systems, are often too slow or rigid to keep up with attackers who can change tactics quickly. The future of network security will likely depend heavily on **artificial intelligence (AI)** and **machine learning (ML)** technologies. These tools can analyze vast amounts of data in real-time, detect patterns of suspicious activity, and automatically respond to attacks before they cause significant harm. Essentially, AI and ML will provide a more dynamic, adaptive approach to threat detection.

**Technological Advances:**

The future of network security will be shaped by rapid advancements in technology, which will push the boundaries of how we protect data. One major area of innovation is **AI-driven firewalls**. Unlike traditional firewalls, which rely on predefined rules to filter traffic, AI-driven firewalls will be able to "learn" from the data they process. They can continuously improve their detection abilities by recognizing new, previously unseen types of attacks and responding to them much faster than a human ever could. Over time, these firewalls will become smarter and more effective in identifying subtle security risks.

Another groundbreaking development on the horizon is **quantum encryption**. Quantum computing has the potential to break current encryption standards by solving complex mathematical problems faster than traditional computers. This poses a huge threat to today's security infrastructure. However, quantum encryption uses the principles of quantum mechanics to create nearly unbreakable encryption. By leveraging quantum particles, it could secure data in ways that are fundamentally different—and vastly more secure—than the encryption methods we use today. In the future, quantum encryption could become a key defense mechanism, especially as quantum computing becomes more widespread.

By getting both together, these advancements signal a future where network security isn't just about building stronger walls but making systems smarter and more resilient. Companies will need to embrace these technologies to stay ahead of increasingly sophisticated cyber threats.

**Conclusion**

As digital connectivity continues to advance, securing network infrastructure has become more important than ever. Organizations are under constant threat from both external cyberattacks and internal vulnerabilities, making a multi-layered security approach essential. Core technologies like firewalls, Virtual Private Networks (VPNs), and Intrusion Detection/Prevention Systems (IDS/IPS) are the foundation of a strong defense strategy, with each playing a unique yet complementary role in keeping networks safe.

Firewalls act as the first layer of protection by controlling the traffic flowing between internal and external networks. They block unauthorized access while allowing legitimate communication, but firewalls alone can’t handle all the sophisticated threats out there today. VPNs come into play by securing data as it travels across public networks, ensuring sensitive information stays private—even when users are working remotely. With remote work becoming more common, VPNs have proven critical in protecting data during transit.

Even with strong firewalls and secure VPNs, there are still risks. This is where IDS and IPS systems become crucial. They actively monitor network traffic for any suspicious activity, alerting administrators (in the case of IDS) or automatically taking action (in the case of IPS) to block harmful traffic. These tools are essential for detecting and stopping advanced threats like malware and unauthorized access attempts in real-time.

A well-rounded network security strategy integrates firewalls, VPNs, and IDS/IPS to create a strong, adaptive defense against the constantly evolving threats organizations face. Firewalls control access, VPNs protect data during transmission, and IDS/IPS identify and block internal threats. Used together, they provide a layered security approach that minimizes vulnerabilities and strengthens an organization’s resilience against cyberattacks.

Looking ahead, it’s clear that organizations must continually adapt their security measures to keep up with the ever-changing landscape of cyber threats. Technologies like artificial intelligence, machine learning, and advanced encryption will play increasingly important roles in strengthening firewalls, VPNs, and IDS/IPS. These innovations will ensure network defenses are capable of tackling the sophisticated threats of the future.

In summary, firewalls, VPNs, and IDS/IPS are fundamental to any robust network security strategy. When used together, they offer organizations the tools they need to protect sensitive data, ensure secure communication, and guard against cyberattacks. By continuously improving and adapting these defenses, businesses can stay ahead of emerging threats and ensure the long-term security of their networks and assets.

**2. Vulnerability Scanning with Nmap**

**Task:**Run an advanced Nmap scan on your local network. Identify open ports, running services, and potential vulnerabilities.

Write a 1page report with your findings, explaining the risks and how they could be mitigated. Mention any new commands or techniques you used compared to Week 1.

**Running an advanced Nmap Scan:**

Steps to run the scan on your local network:

* **Identify your local network range**: First, find your IP address and network range using the ifconfig or ip addr command (Linux/macOS) or ipconfig (Windows).

Example output of ifconfig:

inet 192.168.1.15 netmask 255.255.255.0 broadcast 192.168.1.255

The IP range is typically 192.168.1.0/24 or similar.

* **Advanced Nmap Scan Command**: Run an advanced scan to detect open ports, services, and versions:

sudo nmap -sS -sV -O -A 192.168.1.0/24

Explanation of flags:

* + -sS: Stealth SYN scan (faster and less detectable)
  + -sV: Detect services and versions
  + -O: OS detection
  + -A: Aggressive scan (includes additional info like traceroute and script scanning)

**Findings and Mitigations**

**Scan Summary**

* **Date of Scan**: 16-09-24
* **Network Range Scanned**: 192.168.1.0/24
* **Command Used**: sudo nmap -sS -sV -O -A 192.168.1.0/24

**Findings**  
During the scan, I identified the following open ports and services:

| **IP Address** | **Open Ports** | **Service** | **Version** | **Potential Risks** |
| --- | --- | --- | --- | --- |
| 192.168.1.10 | 80 | HTTP | Apache 2.4.29 | Unpatched web server, subject to vulnerabilities like XSS or DDoS |
| 192.168.1.12 | 22 | SSH | OpenSSH 7.6 | Weak authentication could lead to brute-force attacks |
| 192.168.1.15 | 445 | SMB | Samba 4.3.11 | SMB service susceptible to exploitation, such as EternalBlue attack |
| 192.168.1.18 | 3389 | Remote Desktop (RDP) | Microsoft RDP 10.0 | Could allow unauthorized remote access if improperly configured |

**Risks and Mitigations**

1. **HTTP (Port 80)**:  
   Risk: The web server may have vulnerabilities such as Cross-Site Scripting (XSS) or Distributed Denial of Service (DDoS) attacks.  
   Mitigation: Ensure the server is up to date with security patches, and consider switching to HTTPS.
2. **SSH (Port 22)**:  
   Risk: SSH servers are vulnerable to brute-force attacks if weak authentication methods are used.  
   Mitigation: Enable strong password policies, disable root login over SSH, and use key-based authentication.
3. **SMB (Port 445)**:  
   Risk: SMB vulnerabilities, like EternalBlue, could allow remote code execution.  
   Mitigation: Update the SMB service and disable SMBv1. Use a firewall to restrict access to this port from untrusted networks.
4. **RDP (Port 3389)**:  
   Risk: RDP services can be exploited for unauthorized access.  
   Mitigation: Enforce Network Level Authentication (NLA), use strong passwords, and restrict RDP access using a VPN or firewall rules.

**New Commands & Techniques** Compared to Week 1, I used more advanced scanning techniques like OS detection (-O flag) and service version detection (-sV flag). These allowed me to gather detailed information about the services running on each system, which wasn't part of my initial scan in Week 1.

**3. Introduction to Password Cracking (Optional for Advanced Interns)**

**Task:** Install and explore a simple tool like John the Ripper (JTR).Perform a basic passwordcracking exercise on weak passwords. Write a brief report on the experience and discuss the importance of strong password policies

**Passwords Cracking:**

**Password Cracking** refers to the process of recovering passwords from stored data, often using brute force, dictionary attacks, or more sophisticated methods like rainbow tables. Understanding why password security is critical helps emphasize why modern systems need strong passwords, two-factor authentication (2FA), and encryption.

Once installed, you can use JTR to crack weak passwords. Start with a simple hash, such as a Linux system password hash (located in /etc/shadow on Linux systems), or create your own hash for this task.

1. **Create a Sample Password Hash**:
   * You can use OpenSSL to create a hash:

echo "weakpassword" | openssl passwd -6 -stdin

This will generate a SHA-512 hash of the password “weakpassword”.

1. **Save the Hash to a File**: Save the hash to a file named password.txt:

echo '$6$randomsalt$...' > password.txt

1. **Run John the Ripper**: Execute the following command to crack the password:

john password.txt

1. **View the Results**: After running, JTR will attempt to crack the hash using various techniques. You can check the cracked password by running:

john --show password.txt

**Overview:**

I explored the basics of password cracking using John the Ripper (JTR). The objective was to understand how weak passwords can be easily compromised, and to reinforce the importance of strong password policies.

**Tools used:**

**John the Ripper (JTR)**: A free and open-source password cracking software that utilizes various cracking methods such as dictionary attacks, brute-force attacks, and rule-based attacks.

I created a weak password hash using the SHA-512 algorithm and saved it to a file. Using JTR, I performed a basic cracking operation to recover the password.

* **Password used**: weakpassword
* **Hash generated**: [Sample SHA-512 Hash]

**Commands executed:**

echo "weakpassword" | openssl passwd -6 -stdin

john password.txt

john --show password.txt

**Results**  
JTR successfully cracked the weak password in a matter of seconds. This demonstrates how easily passwords that are too short or simple can be compromised using common cracking tools.

**Importance of Strong Password Policies**  
Weak passwords are one of the most common security vulnerabilities. It’s crucial to enforce strong password policies to mitigate these risks. Some key recommendations include:

* **Password length**: A minimum of 12-16 characters.
* **Complexity**: Use a combination of letters, numbers, and special characters.
* **Password managers**: Encourage the use of password managers to generate and store complex passwords.
* **Two-factor authentication (2FA)**: Use 2FA wherever possible for an additional layer of security.

**Conclusion**  
This exercise emphasized the critical importance of strong password policies. The ease of cracking weak passwords underscores the need for organizations to implement stringent security measures to safeguard sensitive data and prevent unauthorized access.

**4. Introduction to Web Vulnerabilities (OWASP Top 10)**

**Task:** Choose one vulnerability from the OWASP Top 10 (e.g., SQL Injection or XSS). Research a realworld example of the vulnerability being exploited and write a 1page summary of the attack, including how it could have been prevented.

**SQL Injection Vulnerability: Heartland Payment Systems Breach (2008)**

**Overview of SQL Injection (SQLi):**  
SQL Injection is a web vulnerability that allows attackers to manipulate a website’s database by injecting malicious SQL queries through input fields. When websites fail to properly sanitize user inputs, attackers can execute arbitrary SQL code, leading to unauthorized access, data theft, and other serious issues.

**Real-World Example: Heartland Payment Systems Breach**  
In 2008, Heartland Payment Systems, a payment processing company, suffered one of the largest data breaches due to an SQL Injection vulnerability. Hackers exploited weaknesses in Heartland’s website to insert malicious SQL queries, gaining access to the company’s internal databases. As a result, over 130 million credit card numbers were stolen, leading to widespread financial losses and the exposure of sensitive consumer information.

**How the Attack Happened?**  
The attackers took advantage of Heartland’s failure to properly sanitize input fields in its web applications. By injecting SQL code into a vulnerable field, the hackers were able to bypass security controls and execute arbitrary commands. This allowed them to access and exfiltrate sensitive data from the backend database.

**Impact:**

* **Data Compromised**: Over 130 million credit card numbers
* **Financial Damage**: Heartland faced millions in fines, lawsuits, and settlements. The breach severely damaged the company’s reputation and led to increased scrutiny of payment security.

**Prevention Measures:**  
This attack could have been prevented by implementing the following security measures:

1. **Input Validation and Sanitization**: Ensuring that all user inputs are validated and sanitized before being passed to the database would have blocked the malicious SQL queries.
2. **Parameterized Queries (Prepared Statements)**: Using parameterized queries ensures that SQL code and data are separated, preventing attackers from injecting code.
3. **Regular Security Audits**: Conducting routine security audits and penetration testing would have helped identify vulnerabilities before they could be exploited.
4. **Web Application Firewall (WAF)**: A WAF could have detected and blocked malicious SQL Injection attempts by analyzing incoming traffic.

**Conclusion:**  
The Heartland breach underscores the critical importance of protecting web applications from SQL Injection attacks. Proper input validation, prepared statements, and ongoing security audits are essential to prevent similar attacks and safeguard sensitive data.

**Cross-Site Scripting (XSS) Vulnerability: MySpace Worm Attack (2005)**

**Overview of Cross-Site Scripting (XSS):**  
Cross-Site Scripting (XSS) is a web security vulnerability that allows attackers to inject malicious scripts into websites viewed by other users. These scripts can be used to steal sensitive data, hijack user sessions, or redirect users to malicious sites. XSS typically occurs when a website doesn’t properly validate or escape user input.

**Real-World Example: MySpace Worm (Samy Worm) Attack**  
In 2005, MySpace was compromised by a self-propagating XSS attack known as the **Samy Worm**. Samy Kamkar, the attacker, exploited a vulnerability in MySpace’s profile feature, embedding malicious JavaScript into his own profile. When other users visited his profile, the script executed in their browsers and added Samy as a friend to their own profile. The worm spread rapidly, affecting over a million MySpace profiles within 24 hours.

**How the Attack Happened?**  
MySpace failed to sanitize user input on profile pages, allowing Kamkar to inject JavaScript into his profile’s HTML content. The script executed automatically whenever someone viewed his profile, adding the phrase "but most of all, Samy is my hero" to the victim’s page and sending a friend request to Kamkar’s profile. This led to the rapid, viral spread of the worm across the platform.

**Impact:**

* **Affected Users**: Over 1 million MySpace profiles were infected within a day.
* **Reputation Damage**: The attack exposed significant security flaws in MySpace, damaging its credibility as a secure social network.
* **Legal Action**: Samy Kamkar faced legal consequences for his role in the attack, although he didn’t steal any personal data.

**Prevention Measures:**  
This attack could have been prevented by implementing the following security measures:

1. **Input Sanitization**: Properly filtering and sanitizing user inputs to remove any potential malicious scripts would have blocked the XSS exploit.
2. **Output Encoding**: Encoding dynamic data before rendering it in the browser would prevent injected scripts from executing.
3. **Content Security Policy (CSP)**: A strict CSP would have restricted the execution of untrusted JavaScript, reducing the potential damage of an XSS attack.
4. **Use of HTTPOnly and Secure Cookies**: By flagging cookies as HTTPOnly, attackers would not be able to access session tokens via JavaScript, limiting the impact of the attack.

**Conclusion:**  
The MySpace Worm demonstrates the devastating impact of unchecked XSS vulnerabilities. By implementing input sanitization, output encoding, and content security policies, organizations can protect their websites from XSS attacks and safeguard user data and platform integrity.

**5. Capture Network Traffic with Wireshark (Advanced Task)**

**Task:** Capture network traffic using Wireshark (build upon Week 1 tool exploration).Analyze: Filter and analyze the traffic to identify any suspicious activity or common patterns (e.g., DNS requests, HTTP traffic). Write a short report (1 page) on what you learned, including any interesting insights from the captured traffic.

**Wireshark Network Traffic Analysis:**

**Introduction:**

The purpose of this analysis was to capture and examine network traffic using Wireshark in order to identify common patterns, such as DNS requests and HTTP traffic, and detect any potential suspicious activity. Wireshark, a powerful packet analyzer, provides deep insights into network behavior, enabling us to spot unusual traffic that might indicate security concerns.

**Methodology:**

Wireshark was used to capture traffic over a Wi-Fi connection on my local network. To focus on specific types of traffic, I applied several display filters. For DNS analysis, I used the dns filter to isolate queries and responses, while HTTP traffic was filtered using http. Additionally, I explored TCP traffic by using tcp.port == 80 for unencrypted HTTP traffic and tcp.port == 443 for HTTPS traffic.

**Filters:**

**DNS Traffic:**

The majority of DNS requests were to well-known domains, such as Google and CDN services. However, I noticed a pattern of repeated DNS queries to an unfamiliar domain. Upon further inspection, the domain appeared to be related to a third-party application frequently communicating with the server. No malicious behavior was observed, but it indicated a background process running persistently.

**HTTP Traffic:**

HTTP traffic was largely normal, with common GET and POST requests observed during regular browsing. Most traffic occurred over HTTPS, which is typical for secure communication. However, a few unencrypted HTTP requests were observed, which could present a security risk in transmitting sensitive data.

**Suspicious Activity:**

No immediate red flags or obvious malicious activity were identified. However, there was an instance of an unusual IP address receiving repeated requests from a local device. This behavior could indicate background services or potentially unwanted software, but more investigation would be needed to confirm any threat.

**Conclusion and results:**

The analysis demonstrated the importance of monitoring network traffic for unusual patterns. In this case, persistent DNS queries to unfamiliar domains and occasional unencrypted HTTP requests pointed to areas where network security could be improved. Wireshark proved to be an effective tool in detecting such traffic, which could serve as indicators of misconfigurations or potential vulnerabilities.