**CYBER SECURITY - Securing Web Applications Services & Servers**

**1. Explain MAC spoofing and Email spoofing**

**MAC Spoofing**

MAC (Media Access Control) spoofing is the act of changing the MAC address of a network interface to appear as a different device. This can be used for various purposes, such as bypassing network access controls or evading detection.

**Example:** A user changes the MAC address of their laptop from **00:1A:2B:3C:4D:5E** to **00:6A:7B:8C:9D:0E** to gain access to a network that restricts access based on MAC addresses.

**Email Spoofing**

Email spoofing involves sending emails with a forged sender address. This can be used to deceive recipients into believing the email came from a trusted source, often for malicious purposes such as phishing.

**Example:** An attacker sends an email with the sender address **support@bank.com** to trick users into revealing their bank credentials.

**2. Perform practical of MITM tool and social engineering Tool**

**MITM (Man-in-the-Middle) Tool**

A common MITM tool is **Bettercap**. Here’s a basic example of how to use it:

1. **Install Bettercap:**

sudo apt-get install bettercap

1. **Start Bettercap:**

sudo bettercap

1. **Set the Interface:**

set net.probe on

set net.sniff on

1. **Start the Attack:**

net.probe

**Social Engineering Tool**

A popular social engineering tool is **SET (Social-Engineer Toolkit)**. Here’s a basic example of how to use it:

1. **Install SET:**

sudo apt-get install set

1. **Start SET:**

sudo setoolkit

1. **Select an Attack Vector:**

1) Social-Engineering Attacks

2) Website Attack Vectors

3) Infectious Media Generator

4) Create a Payload and Listener

5) Mass Mailer Attack

6) Arduino-Based Attack Vector

7) Wireless Access Point Attack Vector

8) QRCode Generator Attack Vector

9) Powershell Attack Vectors

10) Third Party Modules

1. **Choose an Option:**

1

1. **Select a Method:**

1) Spear-Phishing Attack

2) Website Attack Vector

3) Infectious Media Generator

4) Create a Payload and Listener

5) Mass Mailer Attack

6) Arduino-Based Attack Vector

7) Wireless Access Point Attack Vector

8) QRCode Generator Attack Vector

9) Powershell Attack Vectors

10) Third Party Modules

**3. Explain Kali linux tool SYN Flooding Attack using Metasploit**

**SYN Flooding Attack:** SYN flooding is a type of DoS attack where the attacker sends a large number of SYN (synchronize) requests to a target machine, exhausting its resources and preventing legitimate traffic from being processed.

**Using Metasploit:**

1. **Start Metasploit:**

msfconsole

1. **Search for SYN Flood Module:**

search syn flood

1. **Use the Module:**

use auxiliary/dos/tcp/synflood

1. **Set Target:**

set RHOST <target\_ip>

1. **Run the Attack:**

run

**4. Find online email encryption service**

Some popular online email encryption services include:

* **ProtonMail:** Offers end-to-end encryption for emails.
* **Tutanota:** Another service that provides end-to-end encryption.
* **Hushmail:** Known for its strong security features.

**5. Types of Firewall**

1. **Packet Filtering Firewall:** Inspects packets and allows or blocks them based on rules.
2. **Stateful Inspection Firewall:** Keeps track of active connections and uses state tables to determine whether to allow or block traffic.
3. **Proxy Firewall:** Acts as an intermediary for requests from clients seeking resources from other servers.
4. **Next-Generation Firewall (NGFW):** Combines traditional firewall capabilities with other security features like intrusion prevention, application control, and more.

**6. Explain Evading Firewalls**

Evading firewalls is a common tactic used by attackers to bypass security measures and gain unauthorized access to networks or systems. Firewalls are designed to control incoming and outgoing network traffic based on predetermined security rules. Attackers use various techniques to circumvent these rules, making it essential for security professionals to understand these methods. Here are some common techniques used to evade firewalls:

**1. Port Manipulation**

Attackers can manipulate the port numbers used in their traffic to bypass firewalls that are configured to block specific ports.

* **Example:** Using non-standard ports (e.g., 8080 instead of 80 for HTTP) to avoid detection.

**2. Protocol Obfuscation**

Attackers can encapsulate their malicious traffic within legitimate protocols to avoid detection.

* **Example:** Using HTTP tunneling to encapsulate SSH traffic within HTTP requests.

**3. Fragmentation**

Attackers can split their packets into smaller fragments, making it harder for firewalls to inspect the entire payload.

* **Example:** Using tools like **fragroute** to fragment IP packets.

**4. Polymorphic and Metamorphic Code**

Attackers can use polymorphic or metamorphic malware, which changes its code with each iteration, making it harder for signature-based detection systems to identify.

* **Example:** Using a polymorphic shellcode generator to create different versions of the same malicious payload.

**5. Encryption**

Attackers can encrypt their traffic to make it unreadable by firewalls that inspect packet payloads.

* **Example:** Using SSL/TLS to encrypt traffic, bypassing deep packet inspection (DPI) firewalls.

**6. DNS Tunneling**

Attackers can use DNS queries to tunnel data, bypassing firewalls that block non-DNS traffic.

* **Example:** Using tools like **iodine** to tunnel IP traffic over DNS.

**7. IP Spoofing**

Attackers can spoof their IP addresses to make it appear as though the traffic is coming from a trusted source.

* **Example:** Using tools like **hping3** to send packets with a spoofed source IP address.

**8. Social Engineering**

Attackers can trick users into bypassing firewall restrictions through social engineering techniques.

* **Example:** Phishing emails that trick users into downloading malware or providing credentials.

**9. Exploiting Firewall Misconfigurations**

Attackers can exploit misconfigurations in firewalls to gain unauthorized access.

* **Example:** Finding and exploiting a default or weak password on a firewall management interface.

**10. Using VPNs and Proxies**

Attackers can use VPNs or proxies to route their traffic through different networks, bypassing firewall restrictions.

* **Example:** Using a VPN service to route traffic through a different geographic location.

**11. Exploiting Application Layer Protocols**

Attackers can exploit vulnerabilities in application layer protocols to bypass firewalls.

* **Example:** Using SQL injection to bypass firewall restrictions by exploiting vulnerabilities in web applications.

**12. Using Covert Channels**

Attackers can use covert channels to exfiltrate data through seemingly innocuous traffic.

* **Example:** Embedding data in the padding bits of network packets.

**13. Exploiting Weak Authentication Mechanisms**

Attackers can exploit weak authentication mechanisms to gain unauthorized access.

* **Example:** Using brute-force attacks to crack weak passwords on firewall management interfaces.

**14. Using Backdoors**

Attackers can use backdoors to bypass firewall restrictions by creating hidden entry points.

* **Example:** Installing a backdoor on a compromised system to bypass firewall restrictions.

**15. Exploiting Firewall Rules**

Attackers can exploit firewall rules to gain unauthorized access.

* **Example:** Finding and exploiting a rule that allows traffic from a specific IP address or subnet.

**Tools for Evading Firewalls**

* **Metasploit:** A popular penetration testing framework that includes modules for evading firewalls.
* **Nmap:** A network scanning tool that can be used to map networks and identify open ports.
* **Wireshark:** A network protocol analyzer that can be used to capture and analyze network traffic.
* **Burp Suite:** A web vulnerability scanner that can be used to test web applications for vulnerabilities.
* **Hping3:** A network tool that can be used to craft custom packets and bypass firewalls.
* **dnschef:** A DNS tunneling tool that can be used to tunnel data over DNS.

**Conclusion**

Evading firewalls requires a combination of technical skills and creativity. Attackers use a variety of techniques to bypass firewall restrictions, making it essential for security professionals to stay up-to-date with the latest threats and defenses. By understanding these techniques, security professionals can better protect their networks and systems from unauthorized access

**Web Based Hacking**

* **Session Hijacking:** Session hijacking involves stealing a session token to impersonate a user.

**Techniques:**

**Cookie Theft:** Stealing session cookies.

**Cross-Site Scripting (XSS):** Injecting malicious scripts into web pages.

**Man-in-the-Middle (MITM) Attacks:** Intercepting and modifying traffic between the client and server.

* **DoS/DDoS Attack Tools:**

**LOIC (Low Orbit Ion Cannon):** A popular DoS tool.

**Hping3:** A network tool that can be used to craft custom packets.

**Slowloris:** A tool that performs slow HTTP attacks.

* **SYN Flooding Attack:** SYN flooding involves sending a large number of SYN packets to a target, overwhelming its resources.

**Example:** An attacker sends thousands of SYN packets to a web server, causing it to exhaust its resources and become unavailable to legitimate users.

* **Web App Hacking Methodology:**

**Reconnaissance:** Gathering information about the target.

**Scanning:** Identifying open ports and services.

**Enumeration:** Gathering detailed information about the target.

**Exploitation:** Using vulnerabilities to gain access.

**Post-Exploitation:** Maintaining access and extracting data.

* **SQL Injection Methodology:**

**Identify Injection Points:** Find input fields that interact with the database.

**Craft Payloads:** Create SQL queries that exploit vulnerabilities.

**Execute Payloads:** Send crafted queries to the database.

**Extract Data:** Retrieve sensitive information.

* **SQL Injection with Tool:** Using **sqlmap**:

**Install sqlmap:**

sudo apt-get install sqlmap

**Run sqlmap:**

sqlmap -u "http://example.com/vulnerable.php?id=1" –dbs

* **Difference between VA (Vulnerability Assessment) and PT (Penetration Testing):**

**VA:** Focuses on identifying vulnerabilities without exploiting them.

**PT:** Involves exploiting vulnerabilities to demonstrate potential impacts.

* **Writing a Vulnerability Assessment Report:**

**Executive Summary:** Overview of the assessment.

**Scope:** What was tested.

**Methodology:** Steps taken during the assessment.

**Findings:** Detailed list of vulnerabilities.

**Recommendations:** Suggestions for remediation.

* **Zero Day Attacks:** Zero-day attacks exploit vulnerabilities that are unknown to the vendor and have no available patch.

**Example:** An attacker discovers a vulnerability in a software before the vendor releases a patch, exploiting it to gain unauthorized access

**1. What is Cross-Site Scripting (XSS) and its types?**

**Answer:** Cross-Site Scripting (XSS) is a type of security vulnerability typically found in web applications. It enables attackers to inject malicious scripts into content from otherwise trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.

**Types of XSS:**

1. **Stored XSS (Persistent XSS):** The malicious script is permanently stored on the target server (e.g., in a database, in a comment field, etc.).
2. **Reflected XSS (Non-Persistent XSS):** The malicious script is reflected off a web server, such as in an error message, search result, or any other response that includes some or all of the input provided by the user.
3. **DOM-based XSS:** The vulnerability exists in the client-side code rather than the server-side code.

**2. Explain the OWASP Top 10 Web Application Security Risks.**

**Answer:** The OWASP Top 10 is a regularly updated list of the most critical web application security risks. As of the latest update, the top ten risks are:

1. **Broken Access Control:** Restrictions on what authenticated users are allowed to do are not properly enforced.
2. **Cryptographic Failures:** Sensitive data exposure due to weak or misconfigured cryptography.
3. **Injection:** Execution of unintended commands or code, often SQL or LDAP injections.
4. **Insecure Design:** Inadequate design or architecture of the application.
5. **Security Misconfiguration:** Security settings are not correctly configured.
6. **Vulnerable and Outdated Components:** Use of components with known vulnerabilities.
7. **Identification and Authentication Failures:** Weaknesses in the verification of user identities and authentication mechanisms.
8. **Software and Data Integrity Failures:** Inadequate measures to protect the integrity of software and data.
9. **Security Logging and Monitoring Failures:** Insufficient logging and monitoring, making it difficult to detect breaches.
10. **Server-Side Request Forgery (SSRF):** The ability to induce the server-side application to make requests to an arbitrary domain of the attacker's choosing.

**3. What is Cross-Site Request Forgery (CSRF) and how can it be prevented?**

**Answer:** Cross-Site Request Forgery (CSRF) is an attack that tricks the victim into submitting a malicious request. The attacker exploits the trust that a site has in a user's browser.

**Prevention Techniques:**

1. **Synchronizer Token Pattern (CSRF Tokens):** Include a unique, secret token in each request that the server checks.
2. **SameSite Cookies:** Use the **SameSite** attribute to prevent cookies from being sent with cross-site requests.
3. **Referer Header Check:** Verify that the **Referer** header matches the expected domain.
4. **Custom Header:** Use a custom header to identify legitimate requests.

**4. Explain the difference between SQL Injection and NoSQL Injection.**

**Answer:** **SQL Injection:**

* **Target:** Relational databases like MySQL, PostgreSQL, etc.
* **Mechanism:** Injects malicious SQL code into input fields to manipulate the database.
* **Example:** **SELECT \* FROM users WHERE username = 'admin' --' AND password = '';**

**NoSQL Injection:**

* **Target:** NoSQL databases like MongoDB, Cassandra, etc.
* **Mechanism:** Injects malicious code into input fields to manipulate the database queries.
* **Example:** In MongoDB, an attacker might inject code to bypass authentication: **db.users.find({username: 'admin', password: {$ne: 'password'}})**

**5. What is Insecure Direct Object References (IDOR) and how can it be exploited?**

**Answer:** Insecure Direct Object References (IDOR) occur when an application exposes a reference to an internal implementation object, such as a file, directory, or database key. Attackers can manipulate these references to access unauthorized data.

**Exploitation:**

1. **Guessing Object IDs:** Attackers guess or brute-force object IDs to access unauthorized data.
2. **Parameter Tampering:** Modifying URL parameters to access different objects.
3. **Referer Header Manipulation:** Sending requests with manipulated **Referer** headers to bypass access controls.