# INSTITUTE OF AERONAUTICAL ENGINEERING

## AAT – II (ASSIGNMENT)

**SUBJECT : NETWORK AND WEB SECURITY**

**NAME :DANDU NEHA || ROLL NO : 21951A05B8 || YEAR / SEC : VI / C**

## 1. Explain the difference between DNS spoofing with DNS cache poisoning? Show DNS Cache poisoning working process.

**DNS Spoofing** and **DNS Cache Poisoning** are both techniques used by attackers to manipulate the Domain Name System (DNS), but they are different in terms of their approach and execution.

1. **DNS Spoofing**:
   - DNS Spoofing involves an attacker directly intercepting and altering DNS query responses. The attacker sends forged responses to DNS requests before the legitimate DNS server can respond.
   - This attack typically involves on-path or man-in-the-middle attacks where the attacker is positioned between the user and the DNS server, enabling the interception and alteration of DNS responses.
2. **DNS Cache Poisoning**:
   - Once the malicious information is cached, subsequent requests to the poisoned domain name will return the attacker's forged response.
   - This type of attack can affect many users since the poisoned cache data can be distributed to anyone querying the compromised DNS server.

### DNS Cache Poisoning Working Process

1. **DNS Query Initiation**:
   - A user requests access to a website by typing a URL into their browser.
2. **DNS Query Forwarding**:
   - The user's DNS resolver forwards the request to a DNS server if the information is not already cached.
3. **Attacker Response Injection**:
   - The attacker sends a fake DNS response to the resolver before the legitimate DNS server's response arrives.
4. **Subsequent Requests**:
   - Future requests for the poisoned domain name will return the malicious IP address from the cache, redirecting users to the attacker's site.

## 2. ICMP Internet Control Message Protocol is a protocol used in the Internet Protocol IP suite to provide feedback about network conditions and to diagnose network problems. What are the layers in ICMP?

ICMP is a network layer protocol used for error messages and operational information queries. It operates within the Internet Layer of the TCP/IP model.

**Internet Layer**:

- ICMP operates at the Internet Layer in the TCP/IP model.
- It provides feedback on network conditions and error reporting.
- Common ICMP messages include Echo Request/Reply (used by the `ping` command), Destination Unreachable, and Time Exceeded.

## 3. What are the security protocols and methods of distributed firewall?

A **Distributed Firewall** refers to a security approach where firewall enforcement is distributed across multiple points within a network, rather than centralized. Key protocols and methods include:

1. **Security Protocols**:

   - **IPsec (Internet Protocol Security)**: Encrypts and authenticates IP packets.
   - **TLS/SSL (Transport Layer Security/Secure Sockets Layer)**: Secures data transmission over networks.
   - **SSH (Secure Shell)**: Provides secure remote access and administration.

2. **Methods**:

   - **Host-based Firewalls**: Firewalls installed on individual devices to enforce security policies locally.
   - **Policy Distribution**: Centralized management of security policies which are then distributed to all nodes.
   - **Collaboration between Nodes**: Nodes communicate and collaborate to enforce consistent security policies across the network.

## 4. Network intrusion detection NID is the process of monitoring network traffic for signs of unauthorized access, malicious activity, and other security threats . Describe the Network Intrusion Detection system?

A **Network Intrusion Detection System (NIDS)** is a security solution that monitors network traffic for suspicious activity and potential threats. It analyzes incoming and outgoing network packets to identify anomalies, misuse, and malicious behavior.

1. **Components of NIDS**:
   - **Network Sensors**: Devices placed at strategic points in the network to capture and analyze traffic.
   - **Management Console**: Interface for administrators to monitor and manage the NIDS.
   - **Database**: Stores known attack signatures and anomaly patterns for comparison.
2. **Detection Techniques**:
   - **Signature-based Detection**: Compares traffic against a database of known attack signatures.
   - **Anomaly-based Detection**: Establishes a baseline of normal traffic patterns and flags deviations from this baseline.
   - **Heuristic-based Detection**: Uses algorithms to identify potential threats based on behavior analysis.
3. **Advantages**:
   - Provides real-time monitoring and alerts.
   - Enhances overall network security by providing visibility into network activity.

By implementing and managing these security measures, organizations can better protect their networks from unauthorized access, data breaches, and other cyber threats.

5. **A computer virus is a type of malicious software program that can replicate itself and infect a computer system without the knowledge or permission of the user . When should the computer virus be used? And also explain Impact of computer virus on computer.**

   Computer viruses should never be used as they are illegal and malicious.

   **Impact on a Computer:**

   - **Performance Degradation**: Slows down or crashes the system.
   - **Data Corruption and Loss**: Deletes or corrupts files.
   - **Security Breaches**: Allows unauthorized access.
   - **Network Disruption**: Spreads to other systems causing network issues.
   - **Financial Loss**: Costs for recovery and enhanced security.
   - **Reputation Damage**: Loss of customer trust and business opportunities.

6. **Return oriented programming ROP is a technique used by attackers to execute malicious code on a computer system by leveraging existing code in the systems memory Explain the main idea behind ROP attack with a real time example.**

   **Return Oriented Programming (ROP) Attack**

   **Main Idea:**ROP uses existing code sequences ("gadgets") in the system's memory to execute arbitrary code by manipulating the control flow.

   **Example:**

   1. **Vulnerable Program**: Has a buffer overflow vulnerability.
   2. **Gadget Identification**: Attacker finds gadgets within the executable.
   3. **Payload Construction**: Builds a payload with gadget addresses.
   4. **Exploit Execution**: Overwrites return address to execute the payload.

7. **HTML HyperText Markup Language is the most basic building block of the Web. How it defines the structure of web content.**

   **HTML: Defining the Structure of Web Content**

1. **Elements and Tags**: Tags like `<h1>`, `<p>`, `<a>` create headings, paragraphs, and links.
2. **Document Structure**: Includes `<!DOCTYPE html>`, `<html>`, `<head>`, and `<body>`.
3. **Nested Elements**: Elements can be nested, creating a hierarchy.
4. **Attributes**: Tags can have attributes for additional information.

8. **SQL injection is a type of web application security vulnerability that allows an attacker to interfere with the queries that an application makes to its database . What are the measures that can be taken to prevent SQL injection attacks on a computer system? Explicit it.**

### SQL Injection

SQL injection is a web application security vulnerability that occurs when an attacker inserts malicious SQL code into a query. This allows the attacker to interfere with the application's database queries, potentially leading to unauthorized data access, data manipulation, or even database compromise.

- **User Input Exploitation**: Attackers exploit input fields, such as login forms or search boxes, by entering specially crafted SQL statements.
- **Query Manipulation**: The malicious input is included in the SQL query, altering its execution.
- **Potential Consequences**: This can lead to data breaches, data loss, administrative access to the database, and other harmful outcomes.

To effectively prevent SQL injection attacks, various measures can be implemented. One crucial technique is the use of parameterized queries, which ensure that user inputs are treated strictly as data rather than executable code. For instance, in Python with SQLite, this can be achieved using cursor.execute("SELECT * FROM users WHERE username = ?", (username,)).

Another effective measure is employing stored procedures, where predefined database procedures handle SQL execution, thus separating the code from user inputs. An example in SQL Server is: CREATE PROCEDURE GetUser @Username NVARCHAR(50) AS BEGIN SELECT * FROM users WHERE username = @Username END.

Additionally, input validation and sanitization are vital for ensuring that user inputs adhere to expected formats, thereby reducing the risk of malicious data being processed. Properly escaping inputs, such as using mysqli_real_escape_string in PHP to escape special characters, further protects against injection attacks. Adhering to the principle of least privilege involves limiting database user permissions to the minimum necessary, thereby minimizing the potential impact of a successful injection attack.

## 9. Attacks on user interface in web security refer to the techniques used by attackers to exploit vulnerabilities in the user interface of a website or web application . What are the techniques used by hackers to exploiting attacks into targeted system. Explain it briefly.

**Clickjacking** involves tricking users into clicking on hidden or disguised elements by layering or framing malicious content over legitimate sites. This can lead to unintended actions, such as unauthorized transactions or account changes, and is prevented by using `X-Frame-Options` headers to block embedding in iframes.

**Cross-Site Scripting (XSS)** is another technique where attackers inject malicious scripts into web pages that other users view. This can steal session cookies, redirect users, or manipulate page content, leading to data theft or unauthorized access. Preventing XSS involves sanitizing and escaping user inputs and employing Content Security Policy (CSP) to restrict the types of content that can be loaded.

**Phishing** attacks create fake websites or forms that mimic legitimate ones to trick users into entering sensitive information, such as passwords or credit card details. This can result in credential theft and financial loss. Prevention strategies include educating users about phishing threats, using HTTPS for secure communications, and implementing multi-factor authentication.

**UI Redressing (UI Spoofing)** involves creating deceptive interfaces that appear legitimate but are designed to capture user inputs or mislead users. This can result in the unintended sharing of

sensitive information or actions performed based on false pretenses. Preventing UI redressing involves using framebusting techniques and ensuring the integrity of your UI to prevent unauthorized embedding.

Finally, **Form Hijacking** involves intercepting or altering form submissions to capture data or redirect actions to unauthorized endpoints. This can lead to data breaches or unauthorized changes. To prevent form hijacking, use HTTPS to encrypt data in transit, perform server-side validation, and verify form actions on the server. Implementing these preventive measures helps secure the user interface against various attack vectors.

## 10. Injection flaws refer to a class of security vulnerabilities where an attacker is able to inject and execute unintended code or data into an application . Compare different types of injection flaws with each other and also explain the prevention of injection flaws.

1. **SQL Injection**:
   - **Description**: Malicious SQL code is injected into a query, allowing attackers to manipulate or access the database.
   - **Prevention**: Use parameterized queries, stored procedures, and input validation.
2. **Cross-Site Scripting (XSS)**:
   - **Description**: Malicious scripts are injected into web pages, which can be executed in the context of another user's browser.
   - **Prevention**: Sanitize and escape user inputs, and use Content Security Policy (CSP).
3. **Command Injection**:
   - **Description**: Attackers inject arbitrary commands into an application that is executed by the operating system.
   - **Prevention**: Avoid using user inputs in system commands, validate and sanitize inputs.
4. **LDAP Injection**:
   - **Description**: Malicious LDAP queries are injected, altering directory services queries and potentially accessing unauthorized data.
   - **Prevention**: Use parameterized queries and validate LDAP inputs.
5. **XML Injection**:
   - **Description**: Malicious XML content is injected into an application, which can manipulate XML data processing.
   - **Prevention**: Validate and sanitize XML inputs and use secure XML parsers.
6. **Code Injection**:
   - **Description**: Attackers inject executable code into an application, leading to unintended code execution.
   - **Prevention**: Avoid dynamic code execution and validate all inputs.

**General Prevention Measures**:

**Input Validation**: Ensure all inputs meet expected formats and constraints.

**Output Encoding**: Encode outputs to prevent the execution of injected code.

**Use Security Libraries**: Utilize libraries and frameworks that offer built-in protections.

**Regular Security Testing**: Conduct code reviews and penetration testing to find and fix vulnerabilities.