## SESSION HIJACKING

## What is Session Hijacking

Session hijacking refers to the malicious act of taking control of a user’s web session. A session, in the context of web browsing, is a series of interactions between two communication endpoints, sharing a unique session token to ensure continuity and security. It’s a form of attack where a bad actor steals or manipulates the session token to gain unauthorized access to information or services. The hijacking process typically begins when an attacker intercepts this token, which can be likened to a secret handshake between the user and the website. Once in possession of this token, the attacker gains the ability to masquerade as the legitimate user, potentially causing havoc. The methods of interception can vary, ranging from network eavesdropping to sophisticated phishing attacks. With a significant portion of the global population relying on the internet for banking, shopping, and social interactions and the potential impact of session hijacking is substantial. By hijacking a session, attackers can commit fraud, steal identities, and breach confidential communications. Thus, knowledge of session hijacking is a critical step towards safeguarding our online presence.

## Types of Session Hijacking Attacks:

Session hijacking can manifest in various forms, Attacks are generally categorized as either active or passive. Active attacks involve a [cybercriminal](https://www.imperva.com/learn/application-security/cybercrime/) intercepting and using a session token to gain unauthorized access, while passive attacks may involve monitoring and collecting data without immediate use of the intercepted tokens.

### Common Techniques Used by Hijackers:

Among the arsenal of techniques at a hijacker’s disposal, certain methods stand out due to their prevalence and effectiveness.

* Session sniffing: This technique involves monitoring network traffic to capture valid session tokens.
* Cross-site scripting (XSS): Attackers inject malicious scripts into web pages, which then allow them to steal session cookies from unsuspecting users.
* Session fixation: Here, an attacker forces a user to use a specific session ID, which the attacker has already obtained, to compromise the session.

Each technique requires a tailored approach to mitigation, highlighting the need for a multi-layered security strategy.

## The Impact of Session Hijacking:

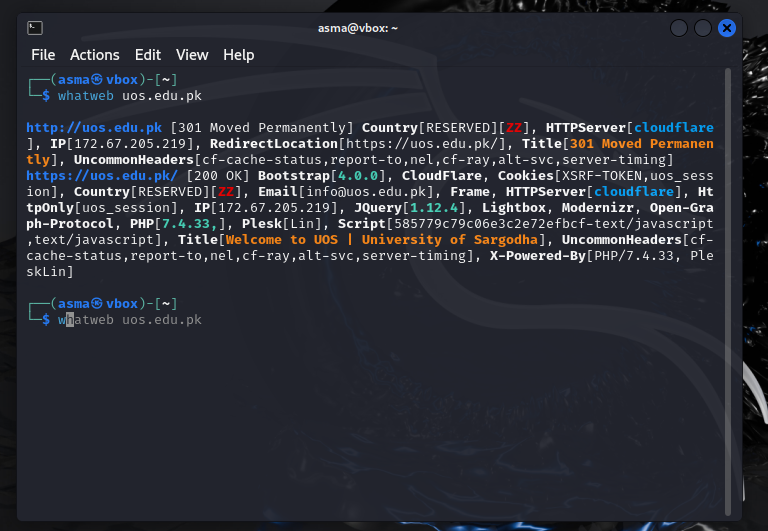
The repercussions of session hijacking are far-reaching, affecting both individuals and organizations. For individuals, a hijacked session can lead to unauthorized access to personal accounts, exposure of sensitive information, and financial loss. The implications for organizations can be even more severe, with the risk of a breach of confidential data, loss of customer trust, and significant financial and reputational damage. At an organizational level, the threat extends beyond immediate losses. A breach of trust can have long-lasting effects on customer loyalty and brand reputation. An organization may even face regulatory fines and legal challenges.

PENETRATION TESTING OF UOS.EDU.PK

1. Tools:

| **Step** | **Task** | **Tool** | **Command/Method** | **Purpose** |
| --- | --- | --- | --- | --- |
| **1** | Domain Information Gathering | whatweb | whatweb uos.edu.pk | Retrieve domain registration details, owner info, and more. |
| **2** | Subdomain Enumeration | Sublist3r | sublist3r -d uos.edu.pk | Enumerate subdomains via search engines. |
|  |  | Amass | amass enum -d uos.edu.pk | Advanced subdomain enumeration and mapping. |
| 3 |  | Wappalyzer | Browser extension | Identify CMS, libraries, and frameworks visually through your browser. |
| **4** | Port and Network Scanning | Nmap | nmap -A uos.edu.pk | Scan open ports and services, identify versions, and gather fingerprints. |
| **5** | Vulnerability Scanning | Nikto | nikto -h uos.edu.pk | Scan for web server vulnerabilities and outdated software. |
| **6** | SSL/TLS Configuration Check | Sslscan | sslscan uos.edu.pk | Analyze SSL/TLS configurations for security compliance. |

1. Information gathering of Uos.edu.pk:



Here is the output of the **WhatWeb** command for **uos.edu.pk**, explained in simple terms with details:

**1. General Information**

* **URL Redirect**:
  + The website initially loads over **HTTP** (http://uos.edu.pk) and then redirects to the secure **HTTPS** version (https://uos.edu.pk).
  + The redirection uses an HTTP status code **301 Moved Permanently**, which means the non-secure HTTP version is permanently redirected to HTTPS for security.
* **IP Address**:
  + The website is hosted on the IP address **172.67.205.219**, which is managed by Cloud flare.
* **Country**:
  + The hosting country is listed as **RESERVED (ZZ)**. This indicates that the specific geographic location of the server is not disclosed by Cloud flare.

**2. Server and Backend Information**

* **Web Server**:
  + The website uses **Cloud flare** as its web server. Cloud flare provides performance optimization and security, such as protection against DDoS attacks.
* **Backend Technology**:
  + **PHP (7.4.33)**: The website is powered by PHP version 7.4.33, which is a server-side scripting language.
  + **Plesk (Lin)**: The server uses Plesk, a popular web hosting management platform.
  + **X-Powered-By Header**: Confirms that PHP 7.4.33 and Plesk are running on the server.

**3. Frontend Information**

* **Bootstrap (4.0.0)**: The website uses the Bootstrap framework (version 4.0.0) for responsive design, ensuring it looks good on both mobile and desktop screens.
* **JavaScript Libraries**:
  + **JQuery (1.12.4)**: A JavaScript library used for interactivity and simplifying tasks like DOM manipulation.
  + **Modernizr**: A library that helps detect browser features and adjust functionality accordingly.
  + **Lightbox**: A JavaScript tool used for creating pop-up or overlay image displays.
* **Open Graph Protocol**:
  + This is used to make the website content shareable on social media platforms (e.g., showing a preview when you share a link).

**4. Cookies and Session Management**

* **Cookies**:
  + **XSRF-TOKEN**: Likely used to prevent **Cross-Site Request Forgery (CSRF)** attacks.
  + **uos\_session**: A session cookie used to identify user sessions. The presence of HttpOnly means it’s inaccessible to JavaScript, improving security.

**5. Email**

* The email **info@uos.edu.pk** is detected in the website's public information. This could be the official contact address.

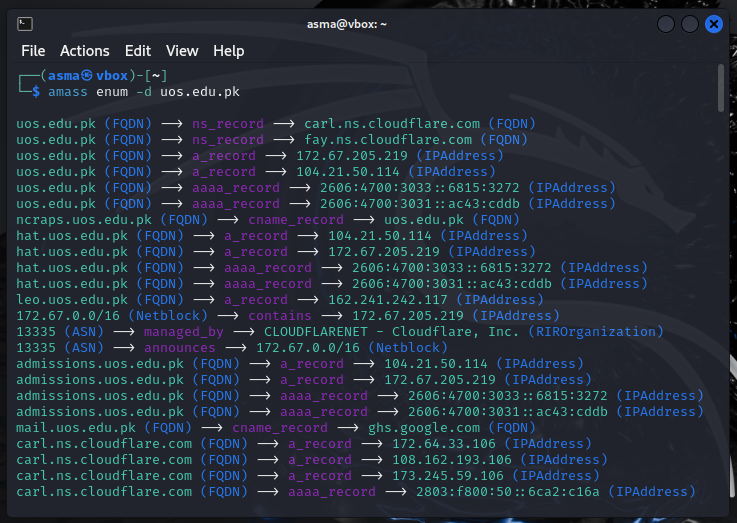
**6. Security and Performance**

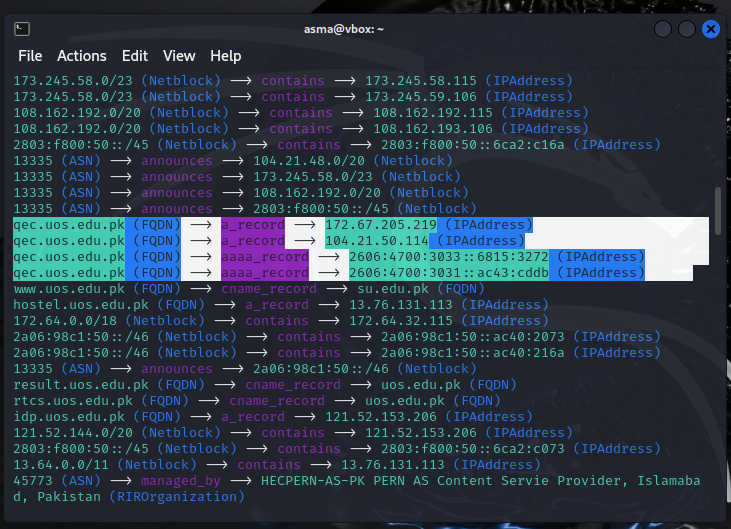
* **Cloud flare**: Acts as a Content Delivery Network (CDN) and security layer to optimize performance and protect the site from malicious traffic.
* **HTTPS**: The use of HTTPS ensures encrypted communication between users and the server.

**7. Miscellaneous**

* **Title**: The page title is "Welcome to UOS | University of Sargodha," which indicates the main purpose of the website.
* **Uncommon Headers**:
  + cf-cache-status, report-to, nel, cf-ray, alt-svc, server-timing:
    - These headers are related to Cloudflare’s performance and security optimizations.

3: Subdomains:





**Primary Domain Records:**

* Nameservers:
  + carl.ns.cloudflare.com
  + fay.ns.cloudflare.com
* IP Addresses:

The two IPv4 addresses, **172.67.205.219** and **104.21.50.114**, are often used for load balancing and redundancy in a content delivery network (CDN) setup, such as Cloud flare. They point to the same website but ensure reliability and efficient traffic distribution, improving performance and fault tolerance.

* + IPv4: 172.67.205.219, 104.21.50.114
  + IPv6: 2606:4700:3033::6815:3272, 2606:4700:3031::ac43:cddb
* Mail Exchange (MX):
  + Multiple Google Mail servers, e.g., aspmx.l.google.com, alt1.aspmx.l.google.com.

Subdomains**:** A variety of subdomains (more than 81) were identified such as:

* admissions.uos.edu.pk
* hostel.su.edu.pk
* qec.su.edu.pk
* webmail.uos.edu.pk
* cpanel.uos.edu.pk
* delivery.uos.edu.pk
* result.uos.edu.pk
* idp.uos.edu.pk (Associated with IP 121.52.153.206 under PERN-AS in Pakistan)
* erozgaar.uos.edu.pk, icpam.uos.edu.pk etc.

Qec.su.edu.pk:

**Website URL**:

* http://qec.su.edu.pk redirects to https://qec.su.edu.pk

The request to the HTTP (non-secure) version of the site results in a **301 Redirect**, meaning the website forces users to the secure HTTPS version.

* The secure HTTPS version of the website responds with a **200 OK**
* The HTTPS version is secure and actively used.

**Frontend Technologies**:

* The website uses **HTML5** for its structure.
* JavaScript libraries used e.g. JQuery and Select2 etc.

**Backend Technologies**:

* The backend is powered by **PHP version 7.4.33**.
* It uses **Plesk** for server management.

**Servers**:

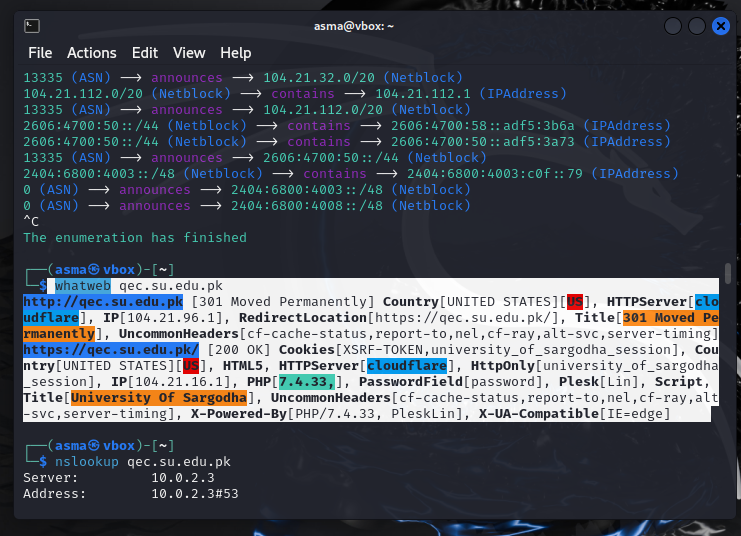
1. **Cloud flare:**
   * Cloud flare is a **reverse proxy** service that provides security, performance optimization, and content delivery.
   * Websites using Cloud flare are protected from common web threats like DDoS attacks, and their content is cached on multiple servers worldwide to improve load times.
   * The HTTP headers and IP addresses in the response indicate that Cloudflare is acting as the front-facing server for this website.
2. **Backend Server:**
   * The website is powered by **PHP 7.4.33** and managed using **PleskLin**, which suggests it is hosted on a Linux server with the Plesk web hosting control panel.
   * Cloudflare sits in front of the actual hosting server, meaning it hides the backend server's details and IP addresses for security and load balancing.

**IP Addresses**:

* 104.21.96.1
* 104.21.16.1
* 104.21.32.1
* 104.21.80.1etc.

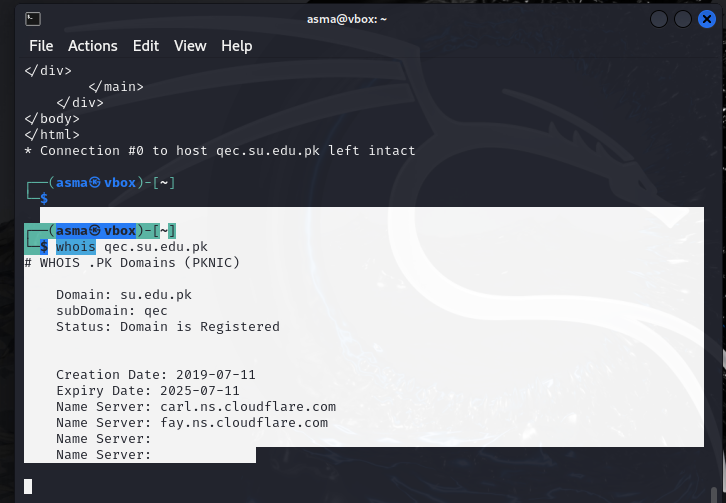
**Security and Cookies**:

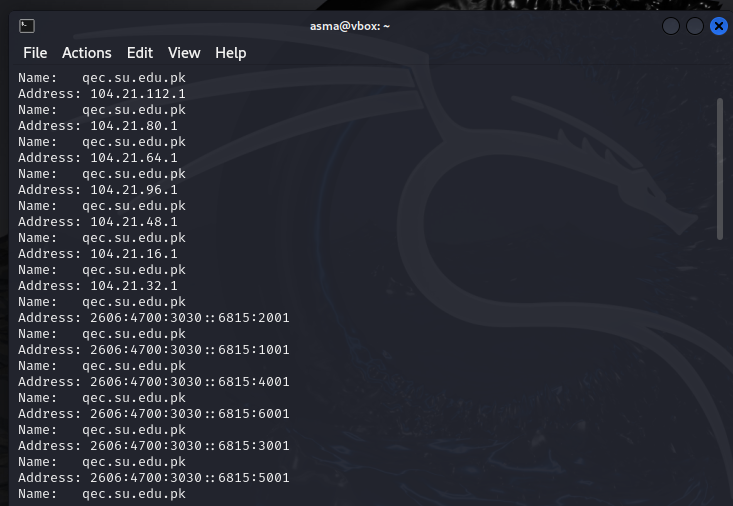
* Uses cookies for session management (e.g., XSRF-TOKEN, university\_of\_sargodha\_session).
* Has a password field on the site (indicating login functionality).



For example:

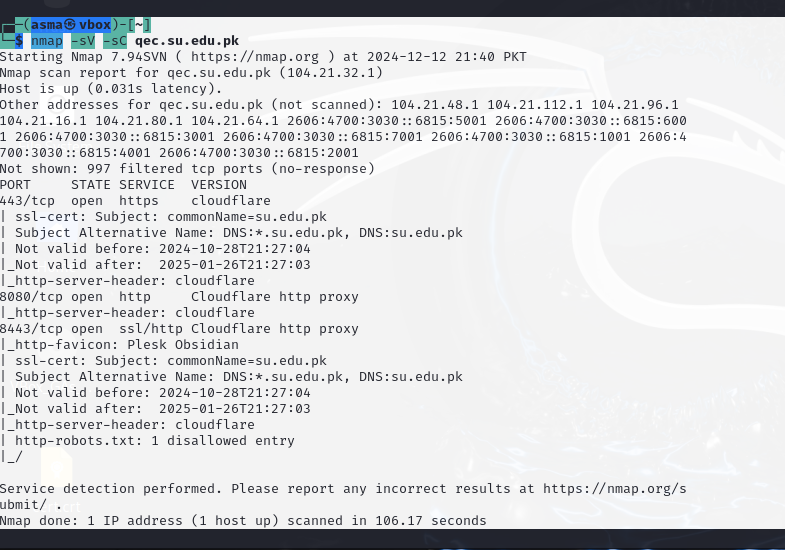
1. You go to http://qec.su.edu.pk.
2. The server tells your browser, "Go to the secure version: https://qec.su.edu.pk."
3. You are connected securely to the HTTPS site, which loads successfully.





4. Vulnerability scanning:

* Vulnerability scanning is conducted to gain initial knowledge and identify any potential security weaknesses that could allow an outside attacker or tester to gain access to the system being tested.
* **Nikto** is an open-source web server scanner used to identify vulnerabilities and security issues in web applications. It is designed to quickly assess a web server for common security flaws, server misconfigurations, and potential vulnerabilities that could be exploited by attackers. Nikto performs a comprehensive scan of a web server and provides valuable information for further security testing or penetration testing.



Here's a simplified explanation of the results from Nmap scan:

**Scan Details**

1. **Target Website**:
   * Domain: qec.su.edu.pk
   * IP Address: 104.21.32.1 (part of Cloudflare's network).
2. **Host Status**:
   * The host is reachable (latency: 0.031 seconds).
3. **Additional IPs**:
   * The domain is associated with multiple IP addresses (e.g., 104.21.48.1, 104.21.112.1, IPv6 addresses, etc.). These represent Cloudflare's global network for load balancing and redundancy.

**Open Ports**

1. **Port 443 (HTTPS)**:
   * **State**: Open.
   * **Service**: Cloudflare-secured HTTPS.
   * **SSL Certificate**:
     + Common Name (CN): su.edu.pk
     + Alternative Names: \*.su.edu.pk, su.edu.pk.
     + Validity:
       - Start Date: October 28, 2024
       - Expiry Date: January 26, 2025
     + This means the SSL certificate is valid and provides encryption for secure connections.
   * **Server Header**: Cloudflare is managing the HTTPS connections.
2. **Port 8080 (HTTP)**:
   * **State**: Open.
   * **Service**: HTTP Proxy via Cloudflare.
   * **Description**: This port is often used for alternate HTTP traffic and is managed by Cloudflare.
3. **Port 8443 (HTTPS)**:
   * **State**: Open.
   * **Service**: HTTPS Proxy via Cloudflare.
   * **Description**: Port 8443 is typically associated with web hosting management tools, like Plesk.
   * **SSL Certificate**: Same as Port 443 (valid and secure).
   * **Favicon**: Indicates Plesk Obsidian, suggesting the backend server might use Plesk for web hosting management.



The Nikto scan https://qec.su.edu.pk produced a range of findings. Here’s a breakdown of each:

### 1. ****Multiple IPs Found****

* **What it means**: Cloudflare uses multiple IPs to manage traffic to the website via its proxy servers. This helps distribute the traffic across several locations, improving performance, load balancing, and redundancy.
* **Purpose**: These multiple IPs ensure that Cloudflare can direct users to the nearest server, enhancing performance and availability.

### 2. ****SSL Info****

* **Subject (CN)**: su.edu.pk - This indicates that the SSL certificate is issued for **su.edu.pk**, providing secure HTTPS communication for the site.
* **Ciphers**: TLS\_AES\_256\_GCM\_SHA384 - This is the encryption method used to secure data between the client and server.

### 3. ****Server Information****

* **Server**: The server is using **Cloudflare** to manage requests, providing enhanced security and performance.
* **X-Powered-By**: The website is running **PHP 7.4.33** and **PleskLin** (suggesting the server is hosted using Plesk, a popular web hosting control panel for Linux).

### 4. ****Security Headers and Potential Issues****

* **Missing Security Headers**:
  + **X-Frame-Options**: This header prevents the site from being embedded in an iframe, reducing the risk of clickjacking attacks. Its absence means the site might be vulnerable to such attacks.
  + **Strict-Transport-Security (HSTS)**: This header enforces the use of HTTPS for all communications with the server, protecting against man-in-the-middle attacks. Its absence means the site is not enforcing strict HTTPS usage.
  + **X-Content-Type-Options**: This header prevents browsers from interpreting files in an unintended manner. Its absence could lead to content type misinterpretation vulnerabilities.
* **Alt-Svc Header**: The site is advertising HTTP/3 support, which is a newer version of HTTP that offers better performance over traditional HTTP/1.1 or HTTP/2, but Nikto cannot test it.

### 5. ****Cookies****

* **XSRF-TOKEN and university\_of\_sargodha\_session Cookies**: These cookies are **missing the secure flag** and **HttpOnly flag**, making them vulnerable to cross-site scripting (XSS) and other types of attacks.
  + **Secure Flag**: This ensures cookies are only sent over HTTPS connections.
  + **HttpOnly Flag**: This helps prevent cookies from being accessed through JavaScript, mitigating the risk of certain attacks.

### 6. ****Robots.txt Issues****

* **Non-forbidden or Redirecting Entries**: The entry /?okparam=/ in robots.txt returned a **200 OK** status, which might allow web crawlers to index sensitive or restricted paths. This could lead to unintended exposure of private resources.
* **Manual Review**: The file contains entries that need to be manually reviewed to ensure there’s nothing that should be hidden from search engines or public access.

### 7. ****SSL/TLS Handshake Errors****

* **SSL Negotiation Failed**: Multiple errors occurred related to the SSL handshake, which is the process where the server and client agree on the encryption method. The errors typically indicate issues with SSL/TLS configurations, potentially causing connections to fail for some clients or applications.

**SSL/TLS Security Rating**

* The scan rates the **least strength** of ciphers used by this server as **C**. This rating suggests that while the server supports strong ciphers (like in TLSv1.2 and TLSv1.3), it also supports weaker ciphers (like in TLSv1.0 and 3DES), which drag down the overall security rating.

### 8. ****Content-Encoding Vulnerability (BREACH)****

* **Deflate Compression**: The server is using the deflate compression algorithm, which may expose the website to a **BREACH** attack. BREACH attacks target HTTPS connections that use compression, allowing attackers to infer sensitive data (such as authentication tokens or passwords) through carefully crafted requests.



The output of your **Wapiti** scan on <https://qec.su.edu.pk>shows several findings and actions taken during the assessment.

* Wapiti is loading various security testing modules such as backup, blind SQL, brute force login, XSS, file, SQL injection, shellshock, and others.
* **Explanation**: Wapiti has a modular structure, and these are the types of vulnerabilities the tool is checking for across the target website.

**7. CSP (Content Security Policy):**

* "CSP is not set"
* **Explanation**: The Content Security Policy (CSP) header is not set on the site. CSP is a security feature to prevent Cross-Site Scripting (XSS) and data injection attacks by specifying allowed content sources. Its absence can make the site more vulnerable.

**8. HTTP Headers Issues:**

* **Missing Headers**:
  + X-Frame-Options: Not set. This header prevents clickjacking by disallowing the page from being embedded in iframes.
  + X-XSS-Protection: Not set. This is a browser feature that can mitigate some forms of XSS attacks.
  + **X-Content-Type-Options**: Not set. This header helps prevent browsers from interpreting files as a different MIME type, which can prevent certain attacks.
  + **Strict-Transport-Security**: Not set. This header enforces HTTPS usage for all subsequent requests, helping prevent man-in-the-middle attacks.
* **Explanation**: These missing headers expose the site to various attacks (clickjacking, XSS, MIME sniffing, etc.). These headers should be configured for better security.

**9. Cookie Flags:**

* **Cookies Found**:
  + **XSRF-TOKEN**: Missing HttpOnly and Secure flags.
  + **university\_of\_sargodha\_session**: Missing HttpOnly and Secure flags.
* **Explanation**: The HttpOnly flag prevents cookies from being accessed by JavaScript, mitigating XSS risks. The Secure flag ensures cookies are only sent over HTTPS, protecting them from interception. Their absence exposes the site to session hijacking and XSS attacks.

**10. Module Executions:**

* Various modules like exec (for command execution), file (for file inclusion), sql (for SQL injection), xss (for cross-site scripting), and others are launched.
* **Explanation**: These modules are running specific tests to find vulnerabilities related to command execution, SQL injection, file inclusion, and XSS.

**11. SSRF (Server-Side Request Forgery) Test:**

* Asking endpoint URL https://wapiti3.ovh/get\_ssrf.php?id=ny7j28 for results.
* **Explanation**: Wapiti tests for SSRF vulnerabilities, where an attacker can manipulate the server to make requests on their behalf, potentially accessing internal systems.

**12. Redirect Test:**

* "Launching module redirect"
* **Explanation**: Wapiti is testing for open redirect vulnerabilities, where the website might incorrectly redirect users to malicious sites.

**14. Permanent XSS Test:**

* "Launching module permanentxss"
* **Explanation**: This test searches for persistent XSSvulnerabilities, where malicious code is permanently stored on the server and can be triggered later by users.

3. Attacks defining:

If the vulnerabilities or security issues mentioned are present, an attacker could potentially perform the following types of attacks:

1. **Clickjacking** (due to missing X-Frame-Options header):
   * Attackers can trick users into interacting with hidden or disguised malicious content on a legitimate website.
2. **Man-in-the-Middle (MITM) Attacks** (due to missing Strict-Transport-Security):
   * Attackers could intercept or modify data sent between the user and the website, stealing sensitive information like login credentials.
3. **Cross-Site Scripting (XSS)** (due to missing headers like X-XSS-Protection and Content-Security-Policy, and insecure cookies):
   * Malicious scripts could be injected into the site, stealing user data, executing actions on behalf of users, or spreading malware.
4. **Session Hijacking** (due to insecure cookie flags - HttpOnly and Secure):
   * Attackers could steal session cookies and impersonate legitimate users to gain unauthorized access.
5. **Data Leakage or Information Disclosure** (due to /robots.txt issues):
   * Sensitive information about website paths or resources could be exposed to attackers.
6. **BREACH Attack** (due to Deflate compression vulnerability):
   * Attackers could infer sensitive data (like passwords or tokens) from encrypted traffic using compression side-channel attacks.
7. **Server-Side Request Forgery (SSRF)**:
   * Attackers could make the server send requests to internal systems, potentially exposing internal network resources or services.
8. **Open Redirect** (due to open redirect vulnerability):

* Attackers could redirect users to malicious sites, potentially for phishing or malware delivery.

1. **Persistent XSS** (due to permanent XSS vulnerabilities).

* Malicious scripts can be permanently stored on the server, and every user who visits the page could be compromised.

These vulnerabilities can lead to severe consequences, including data theft, unauthorized access, and further exploitation of the system.

-----------------------------------------------------------------------------

