**INFORMATION SECURITY**



**Session 2023 - 2027**

**Submitted by:**

Sumaira Hafeez 2023-CS-1

**Supervised by:**

Mam Faiza Iqbal

**Course:**

Information Security

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Contents

[**LINUX FUNDAMENTALS AND TRY HACK ME TASKS** 3](#_Toc191635866)

[**1.** **Basic Linux Command:** 3](#_Toc191635867)

[**2.** **Common Attacks:** 4](#_Toc191635868)

[ **How Attackers Use Social Engineering to Gain Initial Access** 4](#_Toc191635869)

[ **Understandings:** 4](#_Toc191635870)

[**3.** **Networking Basics:** 5](#_Toc191635871)

[ **What is IP address and why it is necessary in networking?** 5](#_Toc191635872)

[ **Difference between TCP and UDP with real world Examples?** 5](#_Toc191635873)

[**4.** **Network Service Analysis:** 5](#_Toc191635874)

[ **Common Network Services and their ports:** 5](#_Toc191635875)

[ **Security Risks associated with Open Ports:** 6](#_Toc191635876)

[**5.HTTP and DNS Security** 6](#_Toc191635877)

[ **How HTTP and HTTPS affects security:** 6](#_Toc191635878)

[ **How attackers exploit DNS poisoning:** 6](#_Toc191635879)

[**6.Web Security:** 7](#_Toc191635880)

[ **Role of Cookies in Web Security?** 7](#_Toc191635881)

[ **How SQL injection bypass authentication?** 8](#_Toc191635882)

[ **Key measures to secure a website:** 9](#_Toc191635883)

[**7.** **Rooms Completion** 10](#_Toc191635884)

# **LINUX FUNDAMENTALS AND TRY HACK ME TASKS**

# **Basic Linux Command:**

* **Listing files in /etc/ directory:**

ls command is used to list files in a provided directory. For that first move to specific directory using cd command. It provided following files.



* **Kernel Version:**

It is the version and release of kernel software within a system. It is found using the command uname –r. The version it provided is **6.11.2-amd64**

****

* **Creating File:**

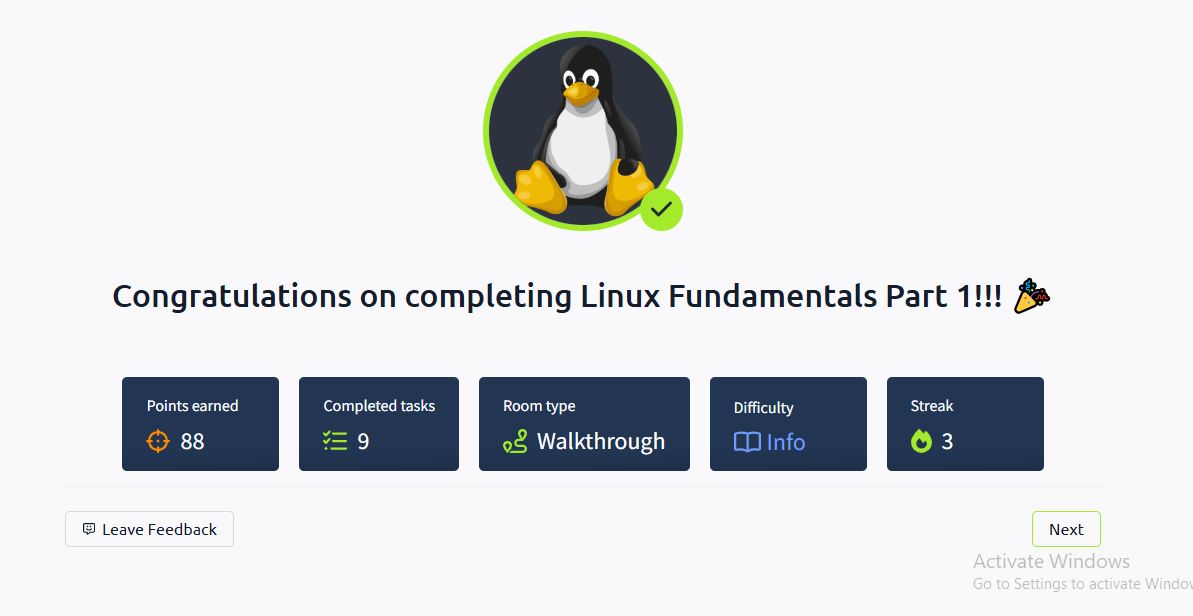
Echo command is used to create a file including the specific description and cat command is used to display the contents.



* **Understandings:**

In this room basic understanding of Linux fundamentals was provided using commands such as listing files using ls command providing changes in a file using echo command. Other important commands described in the room include cd, cat, whoami, uname etc.

* **Completion:**



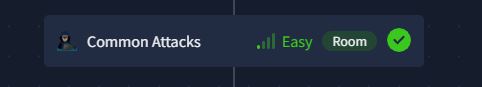
# **Common Attacks:**

### **How Attackers Use Social Engineering to Gain Initial Access**

* **Phishing Emails with Malicious Attachments** – Attackers send emails with infected files that execute malware when opened.
* **Credential Harvesting via Fake Websites** – Users are tricked into entering credentials on a phishing site that looks like a legitimate login page.
* **Impersonation & Urgency Tactics** – Emails appear to come from trusted sources, pressuring users to take immediate action.
* **Malware Delivery via Cloud Services** – Attackers use Google Drive or Dropbox to host malicious files, bypassing security filters.

# **Understandings:**

The room provided details about common attacks in cyber security. It provided description of how social engineering works and how we can save ourselves from such attacks. What is Multi Factor Authentication works and how it can help us in our security and password protection.



# **Networking Basics:**

## **What is IP address and why it is necessary in networking?**

1. **Definition** – An IP (Internet Protocol) address is a unique numerical label assigned to each device in a network to enable communication.
2. **Device Identification** – It helps identify devices on a network, ensuring data is sent to the correct destination.
3. **Facilitates Communication** – IP addresses allow devices to connect and exchange data over the internet or local networks.
4. **Routing & Network Management** – Routers use IP addresses to determine the best path for data packets to reach their destination efficiently.

## **Difference between TCP and UDP with real world Examples?**

1. **Connection Type** – TCP is connection-oriented (ensures data delivery), while UDP is connectionless (sends data without guaranteeing delivery).
2. **Reliability** – TCP ensures reliable data transfer with error checking and retransmission, whereas UDP is faster but does not guarantee delivery.
3. **Speed** – TCP is slower due to its reliability features, while UDP is faster and preferred for real-time applications.
4. **Examples** – TCP is used for web browsing (HTTP/HTTPS) and emails, while UDP is used for online gaming and live streaming, where speed is crucial.

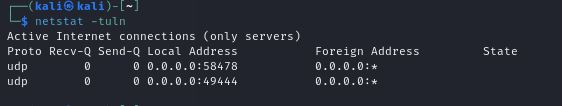


# **Network Service Analysis:**

## **Common Network Services and their ports:**

* 1. HTTP (HyperText Transfer Protocol) uses port 80 is used for accessing the website on internet
  2. HTTPS (HyperText Transfer Protocol Secure) uses port 443 it is the secure version that uses SSL/TLS
  3. DNS (Domain Name System) port 53 it translate domain name into IP address.
* **Checking active services on linux machine:**

netstat –tulnp command is used to check active machines on my linux server. This command has provided two UDP (user Datagram Protocol) services open on my server.



## **Security Risks associated with Open Ports:**

1. **Unauthorized Access**: Attackers can exploit open ports to gain entry into a system, leading to data breaches.
2. **Denial of Service (DoS) Attacks**: Open ports can be flooded with traffic, causing service disruptions.
3. **Malware & Exploits**: Vulnerable services on open ports can be targeted for malware injection or remote code execution.
4. **Data Interception**: Unsecured ports expose sensitive data to interception and man-in-the-middle attacks.

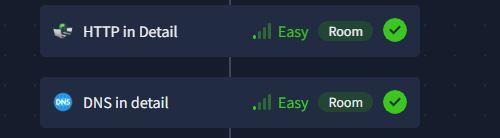
# **5.HTTP and DNS Security**

## **How HTTP and HTTPS affects security:**

1. **Data Encryption**: HTTPS uses SSL/TLS to scramble your data into unreadable code as it travels between your device and the website. This means even if someone intercepts it, they can't understand it. On the other hand, HTTP sends everything in plain text, like writing a postcard—anyone who sees it can read your sensitive information, like passwords or credit card details.
2. **Man-in-the-Middle (MITM) Attacks**: With HTTP, hackers can secretly listen in on your online activity and steal your data. HTTPS, however, creates a secure tunnel for your information, making it nearly impossible for attackers to eavesdrop.
3. **Website Authenticity**: HTTPS uses digital certificates to prove a website is who it claims to be. This helps protect you from fake websites designed to trick you (phishing). HTTP doesn't offer this verification, so you can't be sure if the site you're on is genuine or a scam.
4. **Protection Against Injection Attacks**: HTTPS ensures that the data you send and receive can't be tampered with during transit. With HTTP, attackers can sneakily alter the content, potentially injecting malicious code or stealing your information.

## **How attackers exploit DNS poisoning:**

1. **Spoofing Website Addresses**: Attackers can tamper with DNS servers or caches, tricking them into redirecting users to fake websites. For example, when you type in a legitimate address like "bank.com," the corrupted DNS might send you to a lookalike site like "fakebank.com." These fake sites are designed to mimic the real ones, making it hard to tell the difference.
2. **Credential Theft**: Once you land on a spoofed website, you might unknowingly enter your login details, thinking you're on the real site. Attackers can then steal your username, password, or other sensitive information, giving them access to your accounts.
3. **Malware Distribution**: Poisoned DNS entries can also redirect you to malicious websites that automatically download harmful software, like viruses, spyware, or ransomware, onto your device. This can lead to data loss, device damage, or even complete system lockouts until a ransom is paid.
4. **Traffic Interception**: By manipulating DNS records, attackers can reroute your internet traffic through their own servers. This allows them to monitor, intercept, and steal your data, such as emails, messages, or financial information, without you realizing it.



# **6.Web Security:**

## **Role of Cookies in Web Security?**

**1. User Authentication – Keeping You Logged In**

Imagine you log into your favorite website, and it remembers you so you don’t have to type your password every time you click a new page. That’s cookies at work! They store a small piece of data (like a session token) that tells the website, “Hey, this is still me!” This makes browsing seamless while keeping your login secure—as long as the cookies are properly protected.

**2. Session Management – Keeping Your Data Safe**

Cookies help websites keep track of your session what you’re doing, what’s in your cart, or what settings you’ve chosen. Secure cookies ensure that only the website you’re interacting with can access this information. Without proper session management, attackers could sneak into your session and access your data as if they were you.

**3. CSRF & XSS Protection – Shielding You from Attacks**

Cookies can be a target for attacks like **Cross-Site Request Forgery (CSRF)** and **Cross-Site Scripting (XSS)**. For example:

* In a **CSRF attack**, a malicious site could trick your browser into making unwanted requests using your logged-in session.
* In an **XSS attack**, a hacker could steal your cookies through malicious scripts.

To prevent this, cookies can be configured with special flags:

* **HttpOnly**: Stops scripts (like JavaScript) from accessing the cookie.
* **Secure**: Ensures the cookie is only sent over encrypted HTTPS connections.
* **SameSite**: Limits the cookie to your current site, blocking cross-site misuse.

These flags act like security guards, keeping your cookies safe from prying eyes.

**4. Data Privacy Risks – The Dark Side of Cookies**

When cookies aren’t set up properly, they can become a privacy nightmare. For example:

* Attackers can hijack your session by stealing your cookies, gaining access to your accounts.
* Third-party cookies can track your browsing habits across websites, often without your knowledge.

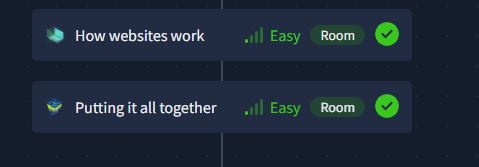
This is why it’s so important for websites to configure cookies securely and for users to be aware of how their data is being used.

## **How SQL injection bypass authentication?**

1. **Injecting Malicious SQL Code**: Attackers enter SQL commands in login fields instead of valid credentials (e.g., admin' OR '1'='1 in the username field).
2. **Always-True Condition**: The injected SQL modifies the query logic to always return true, granting access without a valid password.
3. **Extracting Credential Data**: SQL injection can be used to dump usernames and passwords from the database, allowing attackers to log in as any user.
4. **Altering or Deleting Data**: Attackers can modify or delete authentication records, disrupting system access and security.

## **Key measures to secure a website:**

1. **Implement Input Validation and Sanitization**: Prevents **SQL Injection, XSS, and Command Injection** by validating and sanitizing user inputs.
2. **Use HTTPS and Secure Cookies**: Encrypts data to prevent **MITM attacks**, ensures safe session handling with **Secure, HttpOnly, and SameSite** cookie attributes.
3. **Regularly Update and Patch Software**: Fixes known vulnerabilities by keeping CMS, plugins, and libraries up to date.



# **Rooms Completion**

