**Module 12 - Evading IDS, Firewalls**

**Module 13 – Hacking Web Servers**



**Group Name**

Group 7

**Group Members**

Aayan Rashid – 2023002

Ali Uzair - 2023102

Muaaz Bin Salman – 2023338

**Course**

CY-201

**Instructor**

Sir Abdullah Bin Zarshaid

***Module 12 Lab 1***

**Objective of Module:**

This module is designed to provide hands-on experience in deploying an Intrusion Detection System (IDS) to monitor, analyze, and alert on potential network-based attacks. Participants learn to configure and operate Snort to process real-time traffic, detect anomalies like ICMP ping probes, and interpret the resulting alerts. The lab emphasizes the ability to customize configurations according to specific network environments (e.g., setting the home network IP). The objective is to develop a robust understanding of detecting malicious network activities and to prepare for professional cybersecurity challenges by building practical troubleshooting and analytical skills.

**Environment Setup:**

* Target **Machine (IDS Host):**
  + Windows Server 2019 Virtual Machine
  + IP Address (as configured in Snort): 192.168.162.130
  + Installed tools: Snort v2.9.15-WIN32, WinPcap
* **Attacker Machine:**
  + Windows 11 Virtual Machine (simulating external attack activity)
* **Virtualization Tool:**
  + VMware Workstation
* **Network Settings:**
  + NAT settings to connect with host and to enable internet access
  + Proper routing set up so that the attacker can reach the target
* **Additional Dependencies:**
* Administrative privileges on both machines
* Pre-configured rules and configuration files provided as part of the CEH lab resources

**Tools Used:**

* **Snort (v2.9.15-WIN32):**  
  An open-source IDS that performs real-time traffic analysis, packet logging, and intrusion detection via a customizable ruleset. It monitors network traffic and triggers alerts for suspicious activities (e.g., ICMP ping probes).
* **WinPcap:**  
  A packet capture library for Windows that enables Snort to capture and process network packets. It is vital for ensuring accurate and complete network traffic analysis on the target OS.

**Lab Tasks and Execution Steps**

Below, each activity is documented in detail, including the commands used, screenshots (to be inserted), observed outputs, and an interpretation of the results.

**Task: Detect Intrusions using Snort**

**Objective of Task:**

Deploy and verify Snort on the Windows Server 2019 machine so that it can monitor network traffic and issue alerts (particularly for ICMP-based host discovery/ping probes) when suspicious activity is detected.

**Step 1: Installation & Initial Configuration**

* **Procedure:**
  1. Boot the Windows Server 2019 and Windows 11 VMs.
  2. On Windows 11 VM, navigate to the lab folder  
     Z:\CEHv12 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort  
     and launch the installer file Snort\_2\_9\_15\_Installer.exe.
  3. Follow the installation wizard using default options.
  4. Copy the provided snort.conf and rule directories (i.e., **rules**, **so\_rules**, **preproc\_rules**) to C:\Snort\ as specified in the lab instructions.

**Step 2: Configuring Snort**

* **Procedure:**
  1. Open the file C:\Snort\etc\snort.conf in Notepad++ with administrative rights.
  2. Locate the network variables section and change:
     + var HOME\_NET any to var HOME\_NET 192.168.162.130  
       (Ensure this matches the target machine’s IP where Snort is running.)
  3. Leave EXTERNAL\_NET as “any” and verify that the DNS, SMTP, and other server variables are appropriately set (e.g., DNS\_SERVERS as 8.8.8.8).
  4. Change any relative file paths (for rules, so\_rules, preproc\_rules) to absolute paths:
     + Example: var RULE\_PATH C:\Snort\rules
  5. Optionally comment out or adjust preprocessor configurations that cause “No preprocessors configured for policy 0” warnings when running in IDS mode.
* **Screenshot:**  
  **A close up of a text

  AI-generated content may be incorrect.**

**A black text on a white background

AI-generated content may be incorrect.**

****

****

**Step 3: Network Interface Verification and Snort Execution**

* **Procedure:**
  1. Open a Command Prompt as Administrator.
  2. List available network interfaces using the command:

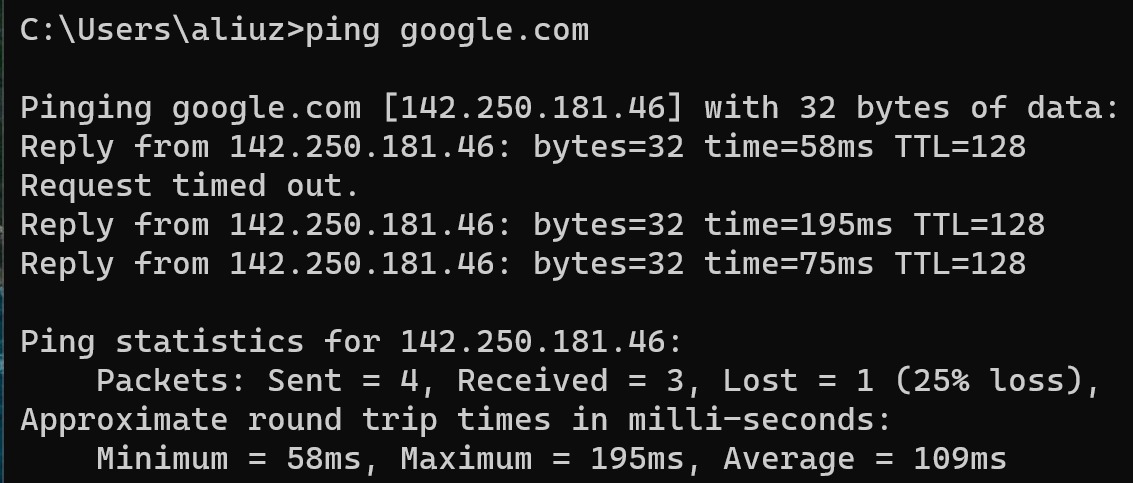
snort -W

* 1. Note down the device index (for example, “1”).
  2. Activate packet capturing on the correct interface with:

snort -dev -i 1

* 1. The output should display real-time packet information indicating that the Ethernet driver is enabled.
  2. In another Command Prompt window, test connectivity by running:

ping google.com

* to observe natural traffic, ensuring that Snort is processing traffic.
* **Screenshot:**  
  ****
* **A screen shot of a computer

  AI-generated content may be incorrect.**

**Step 4: Triggering Intrusion Alerts**

* **Procedure:**
  1. Leave the Snort command prompt window active on the Windows 11 VM machine.
  2. Switch to the Windows 11 host (attacker).
  3. Open the Command Prompt on Windows 11 host and run a continuous ping command directed at the target machine:

ping 192.168.162.130 -t

* 1. This simulates a host discovery attack.
  2. Return to the Windows 11 VM Command Prompt to observe that Snort logs multiple ICMP alerts (e.g., “[1:472:7] ICMP-INFO PING …” messages).
  3. Terminate the ping command and stop Snort (using Ctrl+C) after sufficient alerts have been generated.
* **Screenshot:**  
  **A computer screen with a black screen

  AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Step 5: Verifying Log File Generation**

* **Procedure:**
  1. Navigate to the log directory (e.g., C:\Snort\log\192.168.162.130\ – note that the directory may include the IP of the attacker or appropriate folder name).
  2. Open the log file (e.g., ICMP\_ECHO.ids) using Notepad++.
  3. Verify that multiple alert entries are logged, which confirms that Snort successfully recorded the intrusion events.
* **Screenshot:**  
  **A screenshot of a computer

  AI-generated content may be incorrect.**

**Errors Faced and How They Were Solved**

* **Network Interface Misidentification:**
  + *Issue:* Snort initially failed to capture packets because the wrong interface index was selected.
  + *Solution:* Executed snort -W to list and confirm the correct network interface index (used “1” thereafter with snort -dev -i 1).
* **Preprocessor Configuration Warnings:**
  + *Issue:* Warnings such as “No preprocessors configured for policy 0” appeared on startup.
  + *Solution:* Reviewed and commented out the relevant preprocessor lines in snort.conf that are not essential when running in IDS mode. These warnings do not hinder functionality if the intended IDS operation is maintained.
* **Failed To Create Registry Key:**
* *Issue:* upon running snort, it is unable to create the registry key required to run its services.
* *Solution:* running the command as an administrator resulted in the successful creation of registry key.

**Final Learning and Reflection**

Through this lab exercise, we gained practical experience in configuring a live Intrusion Detection System with Snort. We learned how to tailor the IDS configuration to our specific network environment—setting the correct HOME\_NET, specifying absolute file paths, and managing preprocessor behaviors. The lab reinforced the importance of understanding network interfaces and troubleshooting configuration issues in real time. It taught us about the network connection between different operating systems and gave an insight into the real-time transfer of packets between them. Moreover, the activity enhanced our teamwork and problem-solving skills, as we collaborated closely to isolating issues, test resolutions, and validate our setup. The insights gained here are directly applicable to real-world scenarios where continuous monitoring and rapid response are critical components of cybersecurity defense.

***Module 13 Lab 2***

**1. Objective of the Module**

This module focuses on testing the security of web servers by simulating an FTP credential cracking attack using a dictionary method. The objective is to evaluate how well the FTP service resists unauthorized access attempts via brute-force methods. The lab requires understanding and executing scanning, password guessing, and examining lockout mechanisms. It emphasizes the importance of secure configurations and the risks of using plaintext FTP. The findings illustrate the need for moving toward secure protocols like SFTP and reinforce defensive measures against potential exploits.

**2. Environment Setup**

* **Attacker OS:**  
  Parrot Security OS (Virtual Machine)
* **Target OS:**  
  Windows 11 Virtual Machine running IIS FTP Server
* **Virtualization Tool Used:**  
  VMware / VirtualBox (as applicable)
* **Network Settings:**  
  Bridged networking configuration to allow direct communication between the VMs
* **Additional Dependencies or Tools Installed:**
  + Hydra (v9.5) for dictionary attacks
  + Nmap (v7.94) for network scanning
  + Basic FTP client and command-line utilities
  + **inetmgr (IIS Manager):** Downloaded manually as Windows 11 did not have it pre-installed. This allowed for proper configuration of the FTP service and IP binding.

**3. Tools Used**

| **Tool** | **Version** | **Purpose** |
| --- | --- | --- |
| **Nmap** | v7.94 | Scanning the target for open ports and service identification |
| **THC-Hydra** | v9.5 | Executing the dictionary attack against FTP credentials |
| **FTP Client** | – | Testing FTP connectivity and executing FTP commands |
| **IIS Manager (inetmgr)** | – | Viewing, configuring, and binding the FTP service settings on Windows 11 |
| **Command Prompt/PowerShell** | – | Managing and troubleshooting target user account issues |

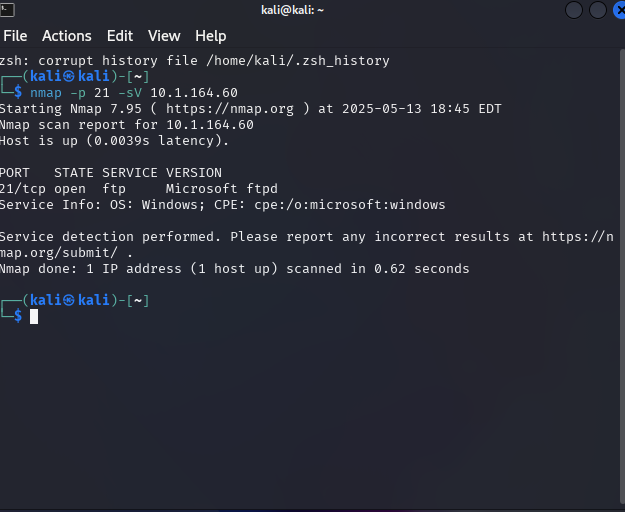
**4. Lab Tasks and Execution Steps**

**Lab Task: Crack FTP Credentials using a Dictionary Attack**

* **Objective of Task:**  
  To use a dictionary attack with THC-Hydra in order to crack FTP credentials on the target Windows FTP server. This helps assess the strength of the FTP account passwords and demonstrates the risks associated with weak credentials.

**Step 1: Scanning the FTP Port**  
**Command:**

nmap -p 21 -sV 10.1.164.60

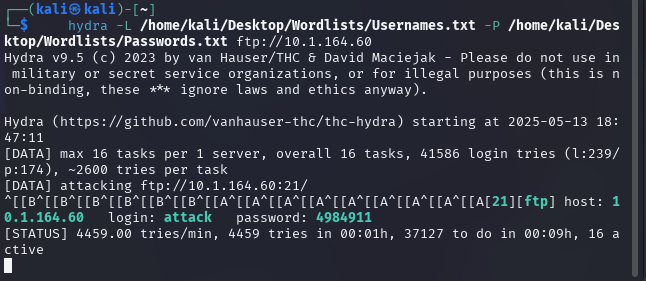
**Screenshot:**  


**Result/Output:**  
The scan confirmed that port 21 is open and provided details about the FTP service version.

**Interpretation:**  
An open port 21 indicates an active FTP service that is potentially vulnerable to brute-force attacks if credentials are weak.

**Step 2: Executing the Dictionary Attack Using Hydra**  
**Command:**

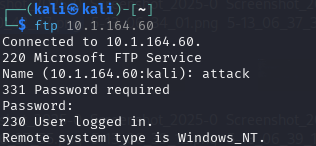
* hydra -L /home/kali/Desktop/Wordlists/Usernames.txt -P /home/kali/Desktop/Wordlists/Passwords.txt ftp://10.1.164.60

**Screenshot:**  
**Result/Output:**  
The dictionary attack successfully **tested multiple username-password combinations** against the FTP server, leading to either successful credential discovery or repeated authentication failures.

**Interpretation:**  
The Hydra command demonstrates how weak FTP credentials can be rapidly tested using automated tools. The response and success of attack attempts help in evaluating the overall security of the FTP service.

**Step 3: Verifying FTP Login with Cracked Credentials**  
**Command:**

ftp[your ip address]

**Screenshot:**  


**Result/Output:**  
The login either succeeded (indicating a successful attack) or was denied (if credentials remained secure).

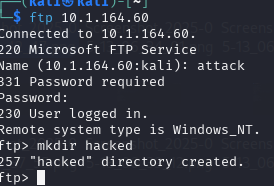
**Interpretation:**  
A successful login validates the vulnerability, while continued failure suggests robust password policies or lockout mechanisms remain in effect.

**Step 4: Creating a File Named "hacked" on the Compromised Device**

Once inside the **compromised FTP system**, a test was conducted to verify **file manipulation permissions** by creating a directory named "hacked" on the target machine.

**Command Used:**

mkdir hacked



**Result/Output:**

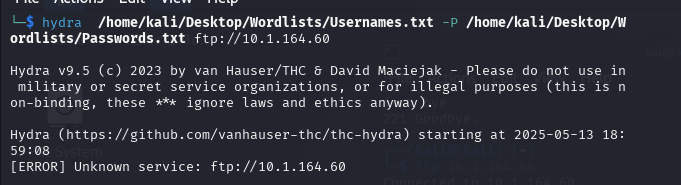
* The "hacked" directory was **successfully created** on the target FTP server, confirming **write access privileges**.

**Interpretation:**

* This test confirms that the cracked FTP credentials **not only allowed access but also permitted unauthorized file manipulation**.
* With **write permissions**, an attacker could potentially:
  + **Upload malicious files**, inject backdoors, or modify system settings.
  + **Delete or replace critical data**, causing operational disruptions.
  + **Escalate privileges** for deeper penetration into the system.

**5. Errors Faced & How They Were Solved**

**Error 1: Hydra Syntax Error**  
**Issue:** Hydra produced an error.  
**Solution:** Reviewed the Hydra command syntax to ensure proper spacing/arguments and corrected the command structure.



**Error 2: File Permission Denied on Wordlists**  
**Issue:** The tool returned permission errors on the wordlist files.  
**Solution:** Changed the permissions using the commands:

* chmod 644 /home/kali/Desktop/Wordlists/Usernames.txt
* chmod 644 /home/kali/Desktop/Wordlists/Passwords.txt

**Error 3: FTP User Account Lockout**  
**Issue:** The account used for the FTP login was locked after multiple failed attempts.  
**Solution:** Reset the lockout using:

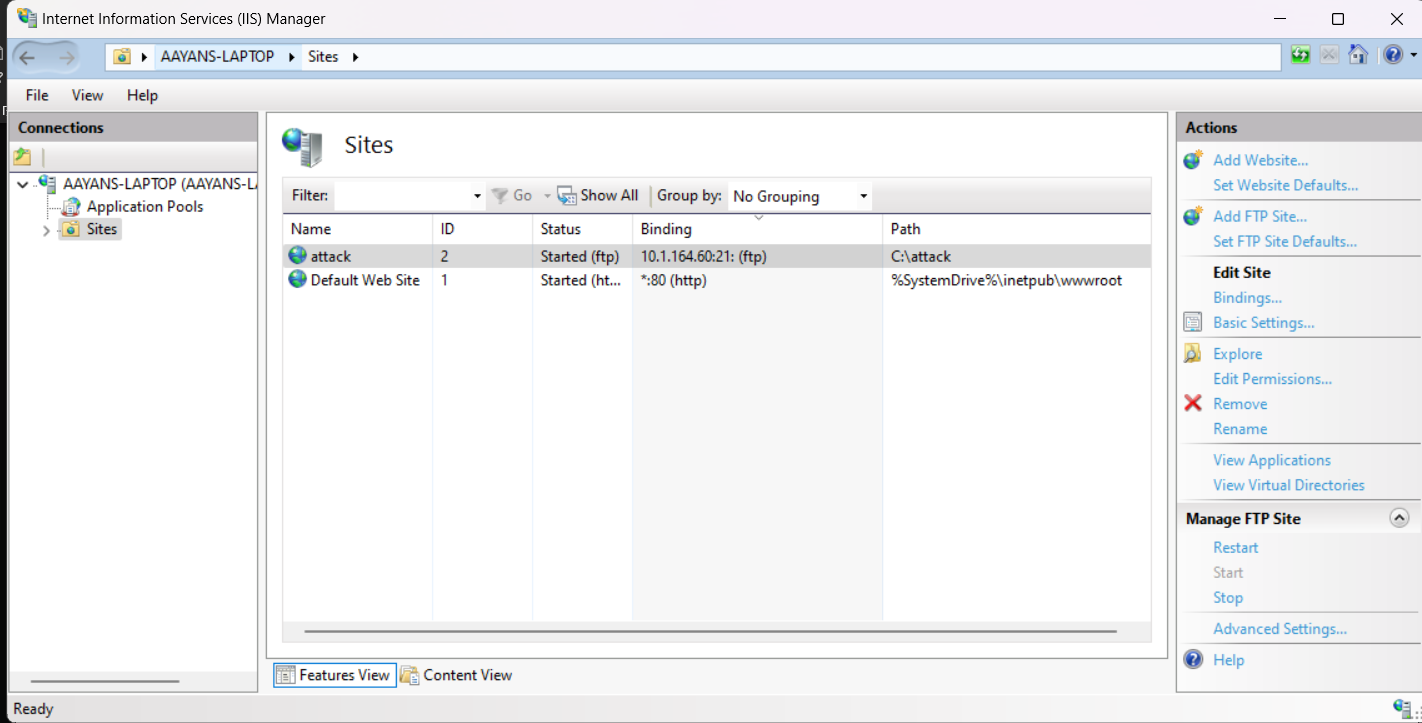
* net user <username> /active:yes

A screen shot of a computer error

AI-generated content may be incorrect.

**Error 4: Configuring FTP on Windows 11**  
**Issue:** Windows 11 did not have inetmgr (IIS Manager) installed by default, and the FTP service was misconfigured with incorrect IP binding, leading to connectivity issues.  
**Solution:**

* + Manually downloaded and installed inetmgr to access IIS Manager.
  + Configured the FTP service by binding it to the specific IP address (10.1.164.60) to ensure proper network interface association.



**6. Potential Exploitation and Risks**

Weak or misconfigured FTP services pose significant security risks. If exploited, attackers could use dictionary or brute-force attacks to gain unauthorized access to the FTP server. This could lead to:

* **Unauthorized Data Access:**  
  Attackers may retrieve sensitive files, documents, and data stored on the server.
* **Server Compromise:**  
  Gaining control of the FTP account could allow an attacker to modify or delete files, inject malicious content, or pivot to other systems within the network.
* **Lateral Movement:**  
  Once inside the network via compromised FTP credentials, attackers might exploit trust relationships to access additional systems or sensitive areas of the network.
* **Reputation and Financial Damage:**  
  A compromised FTP server can damage an organization’s reputation and result in financial losses due to breaches of confidentiality and integrity.

It is imperative to mitigate these risks by enforcing strong passwords, configuring account lockout policies properly, implementing user isolation, and migrating to secure protocols like SFTP which encrypt data in transit.

**7. Final Learning and Reflection**

This lab reinforced the importance of strong password policies and secure FTP configurations. The dictionary attack using Hydra demonstrated how weak credentials can be quickly tested by an adversary. Configuring FTP on Windows 11 presented challenges such as missing utilities and initial misconfigurations—issues that required creating new user accounts, proper IP binding, and manual installation and configuration of inetmgr. Furthermore, the exercise emphasized the serious risks associated with unsecured FTP services, including unauthorized access, data theft, and potential lateral movement within a network. Overall, this experience has underscored the necessity of transitioning to secure protocols like SFTP and improving overall system hardening measures.