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.)

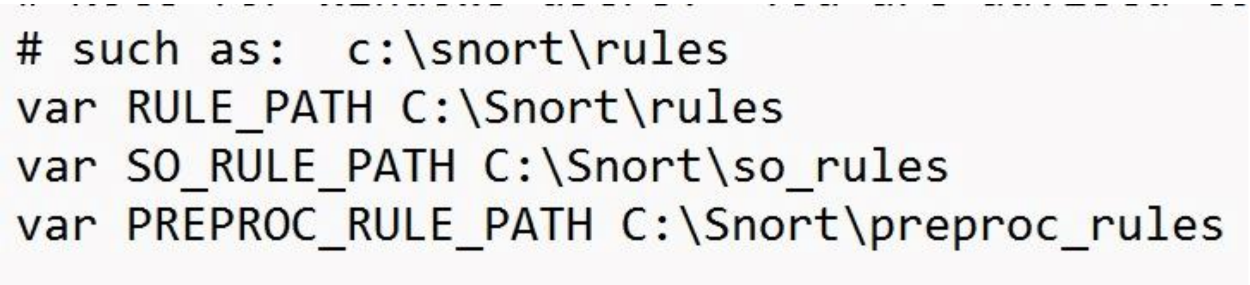
3. Leave EXTERNAL_NET as "any" and verify that the DNS, SMTP, and other server variables are appropriately set (e.g., DNS_SERVERS as 8.8.8.8).

4. Change any relative file paths (for rules, so_rules, preproc_rules) to absolute paths:

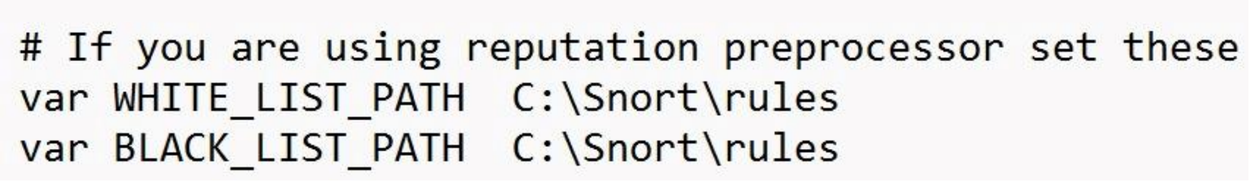
   - Example: var RULE_PATH C:\Snort\rules

5. Optionally comment out or adjust preprocessor configurations that cause "No preprocessors configured for policy 0" warnings when running in IDS mode.

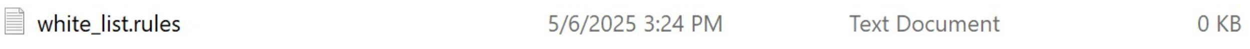
- **Screenshot:**



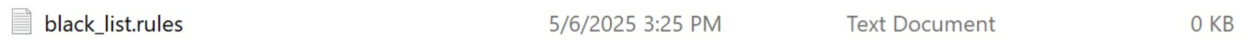
```
# such as:  c:\snort\rules
var RULE_PATH C:\Snort\rules
var SO_RULE_PATH C:\Snort\so_rules
var PREPROC_RULE_PATH C:\Snort\preproc_rules
```



```
# If you are using reputation preprocessor set these
var WHITE_LIST_PATH  C:\Snort\rules
var BLACK_LIST_PATH  C:\Snort\rules
```



white_list.rules 5/6/2025 3:24 PM Text Document 0 KB



black_list.rules 5/6/2025 3:25 PM Text Document 0 KB

<u>Step 3: Network Interface Verification and Snort Execution</u>

- **Procedure:**

1. Open a Command Prompt as Administrator.

2. List available network interfaces using the command:

```
snort -W
```

3. Note down the device index (for example, "1").

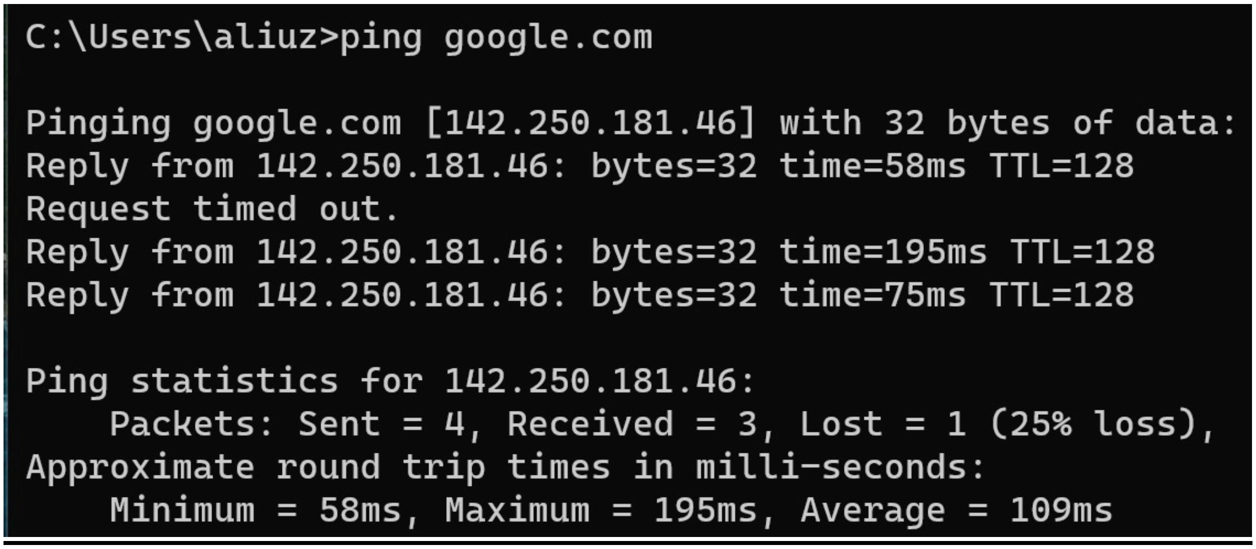
4. Activate packet capturing on the correct interface with:

snort -dev -i 1

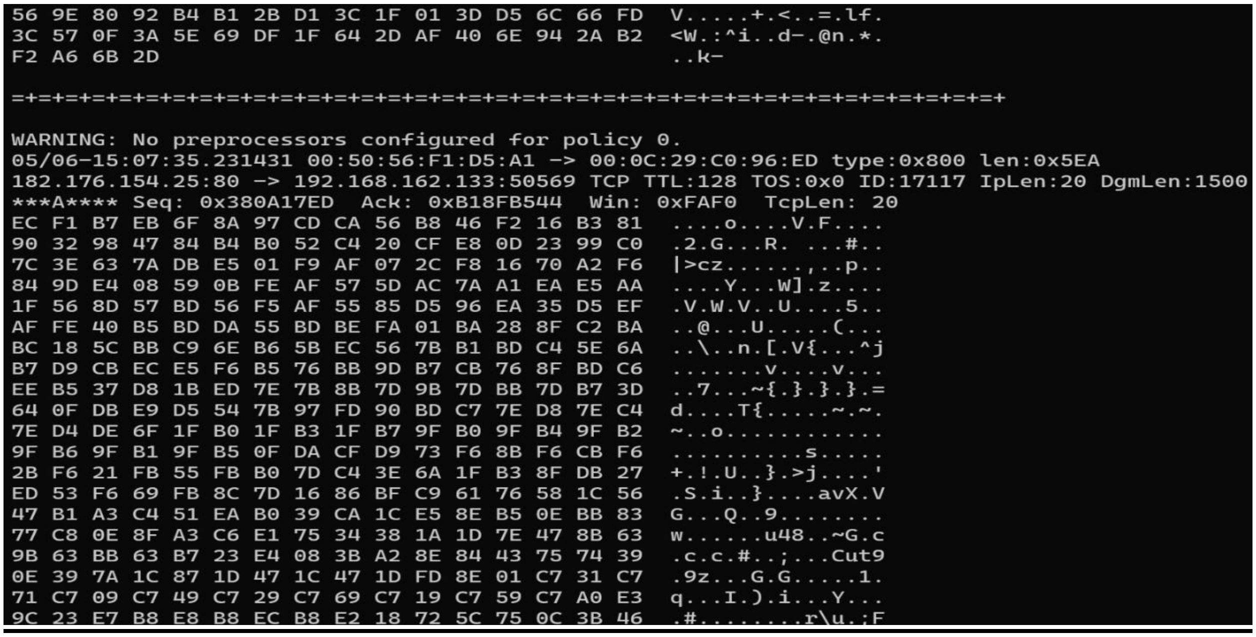
5. The output should display real-time packet information indicating that the Ethernet driver is enabled.
6. In another Command Prompt window, test connectivity by running:

ping google.com

- to observe natural traffic, ensuring that Snort is processing traffic.
- **Screenshot:**



```
C:\Users\aliuz>ping google.com

Pinging google.com [142.250.181.46] with 32 bytes of data:
Reply from 142.250.181.46: bytes=32 time=58ms TTL=128
Request timed out.
Reply from 142.250.181.46: bytes=32 time=195ms TTL=128
Reply from 142.250.181.46: bytes=32 time=75ms TTL=128

Ping statistics for 142.250.181.46:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 58ms, Maximum = 195ms, Average = 109ms
```

- 

```
56 9E 80 92 B4 B1 2B D1 3C 1F 01 3D D5 6C 66 FD  V.....+.<..=.lf.
3C 57 0F 3A 5E 69 DF 1F 64 2D AF 40 6E 94 2A B2  <W.:^i..d-.@n.*.
F2 A6 6B 2D                                      ..k-

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
05/06-15:07:35.231431 00:50:56:F1:D5:A1 -> 00:0C:29:C0:96:ED type:0x800 len:0x5EA
182.176.154.25:80 -> 192.168.162.133:50569 TCP TTL:128 TOS:0x0 ID:17117 IpLen:20 DgmLen:1500
***A**** Seq: 0x380A17ED  Ack: 0xB18FB544  Win: 0xFAF0  TcpLen: 20
EC F1 B7 EB 6F 8A 97 CD CA 56 B8 46 F2 16 B3 81  ....o....V.F....
90 32 98 47 84 B4 B0 52 C4 20 CF E8 0D 23 99 C0  .2.G...R. ...#..
7C 3E 63 7A DB E5 01 F9 AF 07 2C F8 16 70 A2 F6  |>cz......,..p..
84 9D E4 08 59 0B FE AF 57 5D AC 7A A1 EA E5 AA  ....Y...W].z....
1F 56 8D 57 BD 56 F5 AF 55 85 D5 96 EA 35 D5 EF  .V.W.V..U....5..
AF FE 40 B5 BD DA 55 BD BE FA 01 BA 28 8F C2 BA  ..@...U.....(...
BC 18 5C BB C9 6E B6 5B EC 56 7B B1 BD C4 5E 6A  ..\..n.[.V{...^j
B7 D9 CB EC E5 F6 B5 76 BB 9D B7 CB 76 8F BD C6  .......v....v...
EE B5 37 D8 1B ED 7E 7B 8B 7D 9B 7D BB 7D B7 3D  ..7...~{.}.}.}.=
64 0F DB E9 D5 54 7B 97 FD 90 BD C7 7E D8 7E C4  d....T{.....~.~.
7E D4 DE 6F 1F B0 1F B3 1F B7 9F B0 9F B4 9F B2  ~..o............
9F B6 9F B1 9F B5 0F DA CF D9 73 F6 8B F6 CB F6  ..........s.....
2B F6 21 FB 55 FB B0 7D C4 3E 6A 1F B3 8F DB 27  +.!.U..}.>j....'
ED 53 F6 69 FB 8C 7D 16 86 BF C9 61 76 58 1C 56  .S.i..}....avX.V
47 B1 A3 C4 51 EA B0 39 CA 1C E5 8E B5 0E BB 83  G...Q..9........
77 C8 0E 8F A3 C6 E1 75 34 38 1A 1D 7E 47 8B 63  w......u48..~G.c
9B 63 BB 63 B7 23 E4 08 3B A2 8E 84 43 75 74 39  .c.c.#..;...Cut9
0E 39 7A 1C 87 1D 47 1C 47 1D FD 8E 01 C7 31 C7  .9z...G.G.....1.
71 C7 09 C7 49 C7 29 C7 69 C7 19 C7 59 C7 A0 E3  q...I.).i...Y...
9C 23 E7 B8 E8 B8 EC B8 E2 18 72 5C 75 0C 3B 46  .#........r\u.;F
```

Step 4: Triggering Intrusion Alerts

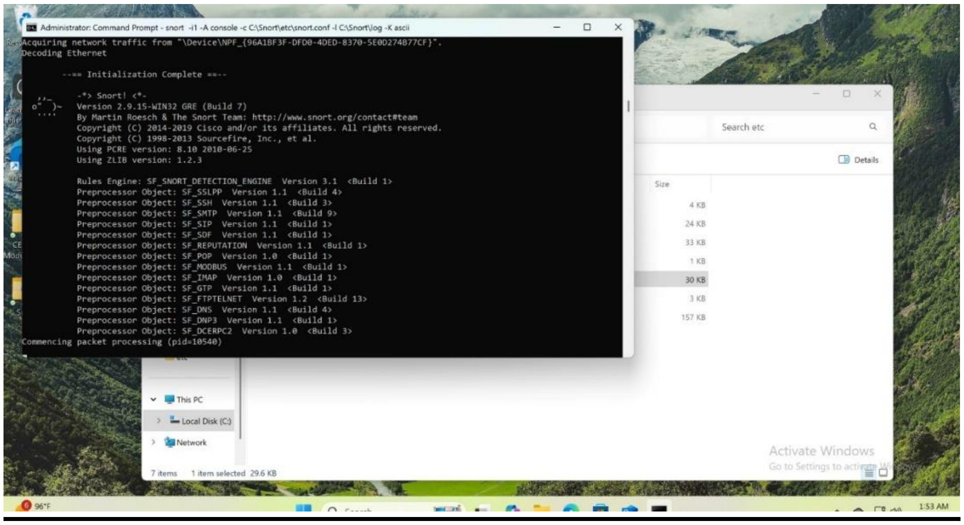
- **Procedure:**

    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:

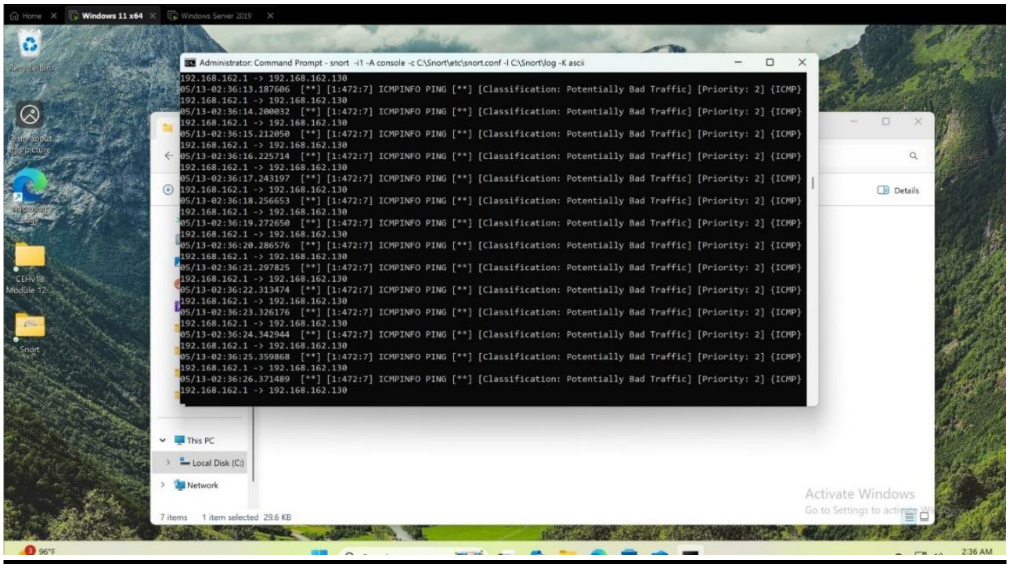
       ping 192.168.162.130 -t

    4. This simulates a host discovery attack.
    5. 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).
    6. Terminate the ping command and stop Snort (using Ctrl+C) after sufficient alerts have been generated.

- **Screenshot:**


Administrator: Command Prompt - snort -i1 -A console -c C:\Snort\etc\snort.conf -l C:\Snort\log -K ascii
Search etc
Details
Size
4 KB
24 KB
33 KB
1 KB
30 KB
3 KB
157 KB
This PC
Local Disk (C:)
Network
7 items 1 item selected 29.6 KB
Activate Windows
Go to Settings to activate Windows.
96°F
1:53 AM

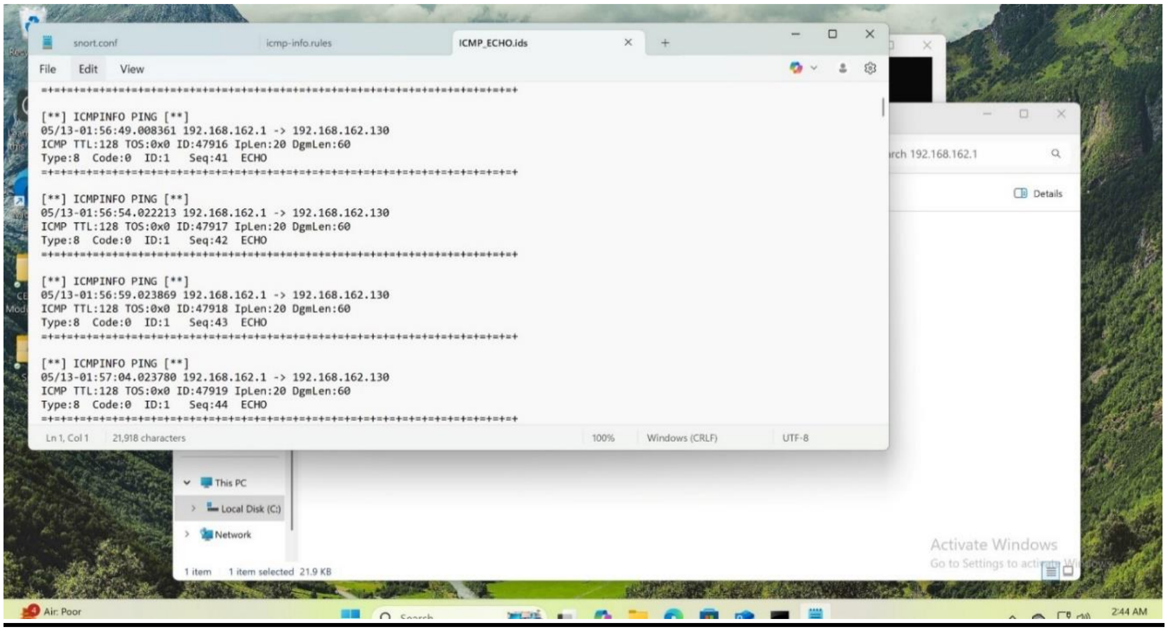
```
Acquiring network traffic from "\Device\NPF_{96A1BF3F-DFD0-4DED-8370-5E0D274B77CF}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15-WIN32 GRE (Build 7)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.3

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
Commencing packet processing (pid=10540)
```


Windows 11 x64
Windows Server 2019
Administrator: Command Prompt - snort -i1 -A console -c C:\Snort\etc\snort.conf -l C:\Snort\log -K ascii
This PC
Local Disk (C:)
Network
7 items 1 item selected 29.6 KB
Details
Activate Windows
Go to Settings to activate Windows.
2:36 AM

```
192.168.162.1 -> 192.168.162.130
05/13-02:36:13.187606  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:14.200032  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:15.212050  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:16.225714  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:17.243197  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:18.256653  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:19.272650  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:20.286576  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:21.297825  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:22.313474  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:23.326176  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:24.342944  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:25.359868  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
05/13-02:36:26.371489  [**] [1:472:7] ICMPINFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP}
192.168.162.1 -> 192.168.162.130
```

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:**



```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] ICMPINFO PING [**]
05/13-01:56:49.008361 192.168.162.1 -> 192.168.162.130
ICMP TTL:128 TOS:0x0 ID:47916 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:41  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] ICMPINFO PING [**]
05/13-01:56:54.022213 192.168.162.1 -> 192.168.162.130
ICMP TTL:128 TOS:0x0 ID:47917 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:42  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] ICMPINFO PING [**]
05/13-01:56:59.023869 192.168.162.1 -> 192.168.162.130
ICMP TTL:128 TOS:0x0 ID:47918 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:43  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] ICMPINFO PING [**]
05/13-01:57:04.023780 192.168.162.1 -> 192.168.162.130
ICMP TTL:128 TOS:0x0 ID:47919 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:44  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

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

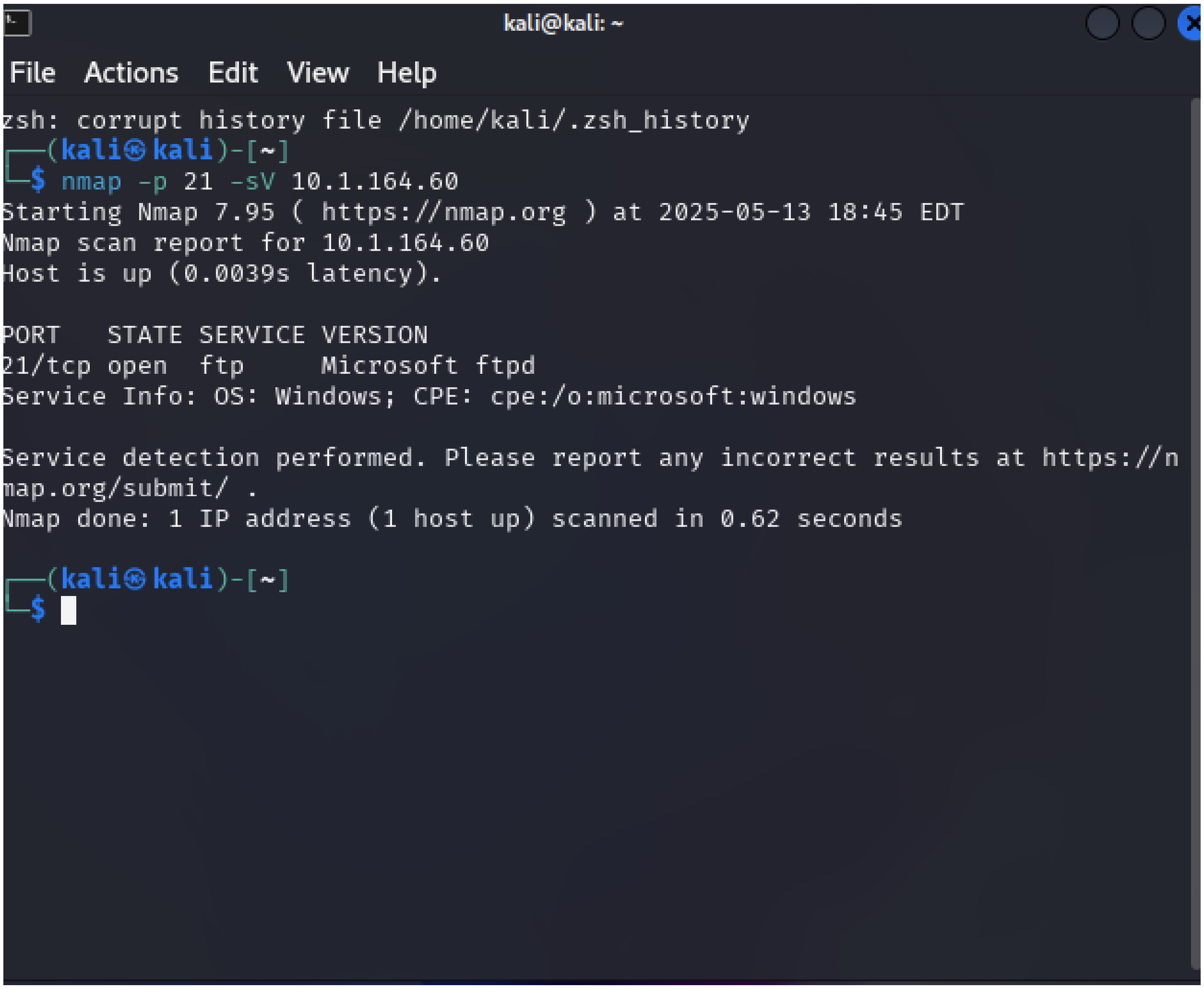
Lab Task: Crack FTP Credentials using a Dictionary Attack

- **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:



```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ nmap -p 21 -sV 10.1.164.60
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-13 18:45 EDT
Nmap scan report for 10.1.164.60
Host is up (0.0039s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.62 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

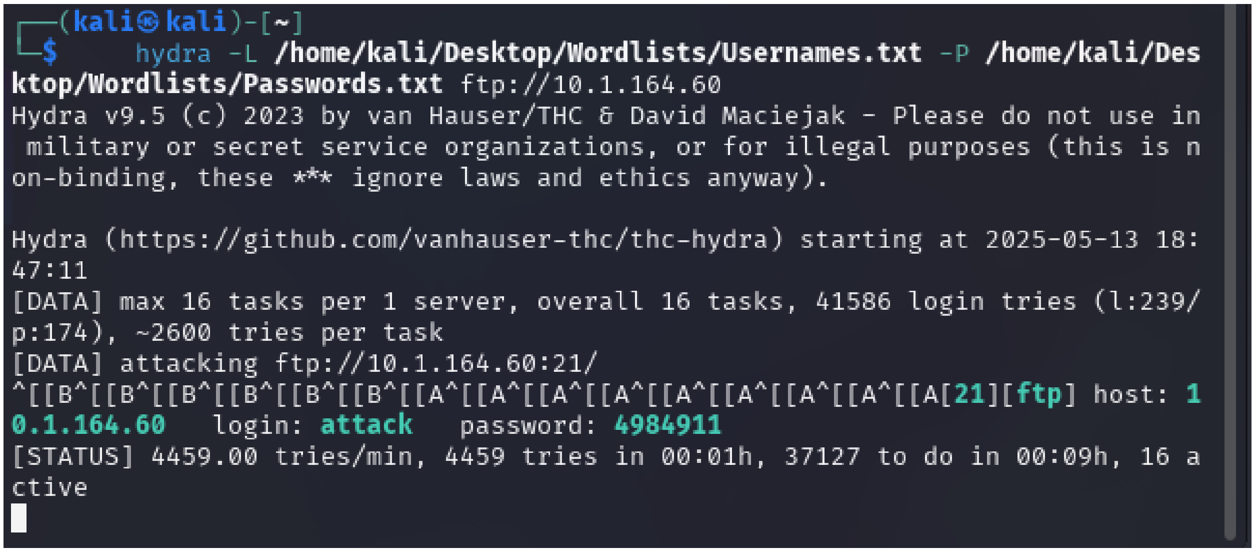
Result/Output:
The scan confirmed that port 21 is open and provided details about the FTP service version.

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:



```
┌──(kali㉿kali)-[~]
└─$     hydra -L /home/kali/Desktop/Wordlists/Usernames.txt -P /home/kali/Des
ktop/Wordlists/Passwords.txt ftp://10.1.164.60
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-13 18:
47:11
[DATA] max 16 tasks per 1 server, overall 16 tasks, 41586 login tries (l:239/
p:174), ~2600 tries per task
[DATA] attacking ftp://10.1.164.60:21/
^[[B^[[B^[[B^[[B^[[B^[[B^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A[21][ftp] host: 1
0.1.164.60   login: attack   password: 4984911
[STATUS] 4459.00 tries/min, 4459 tries in 00:01h, 37127 to do in 00:09h, 16 a
ctive
█
```

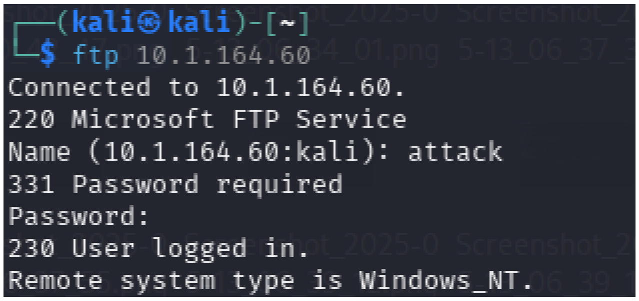
Result/Output:
The dictionary attack successfully **tested multiple username-password combinations** against the FTP server, leading to either successful credential discovery or repeated authentication failures.

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:



```
┌──(kali㉿kali)-[~]
└─$ ftp 10.1.164.60
Connected to 10.1.164.60.
220 Microsoft FTP Service
Name (10.1.164.60:kali): attack
331 Password required
Password:
230 User logged in.
Remote system type is Windows_NT.
```

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

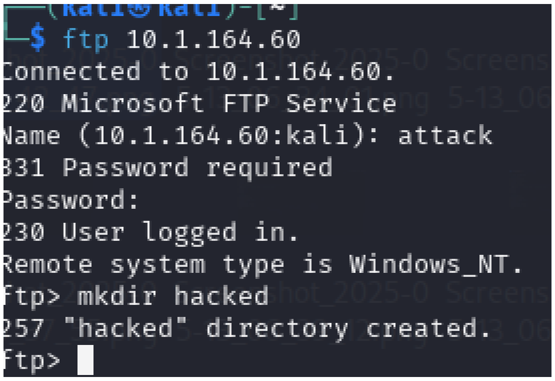
Interpretation:
A successful login validates the vulnerability, while continued failure suggests robust password policies or lockout mechanisms remain in effect.

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



```
┌──(kali㉿kali)-[~]
└─$ ftp 10.1.164.60
Connected to 10.1.164.60.
220 Microsoft FTP Service
Name (10.1.164.60:kali): attack
331 Password required
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> mkdir hacked
257 "hacked" directory created.
ftp> █
```

Result/Output:

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

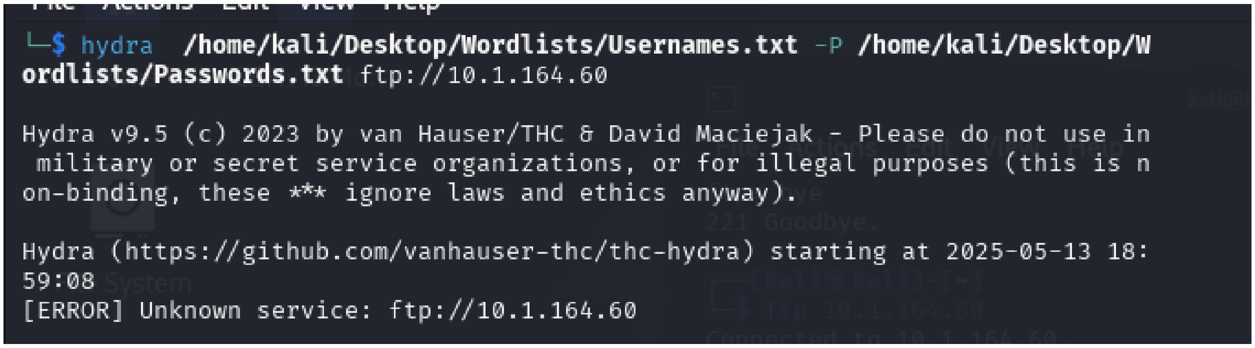
Interpretation:

- 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.



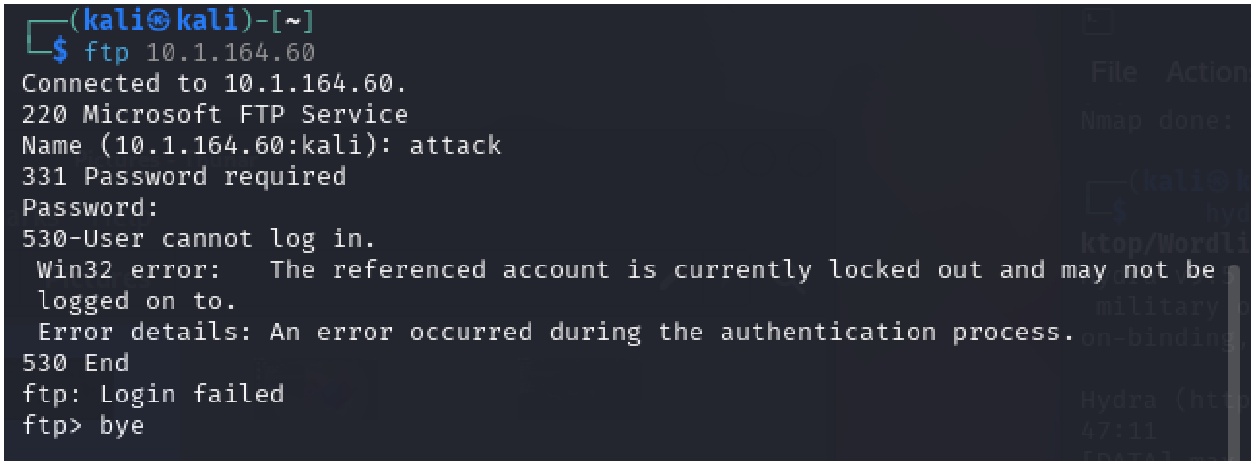
```
└─$ hydra  /home/kali/Desktop/Wordlists/Usernames.txt -P /home/kali/Desktop/W
ordlists/Passwords.txt ftp://10.1.164.60

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-13 18:
59:08
[ERROR] Unknown service: ftp://10.1.164.60
```

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:

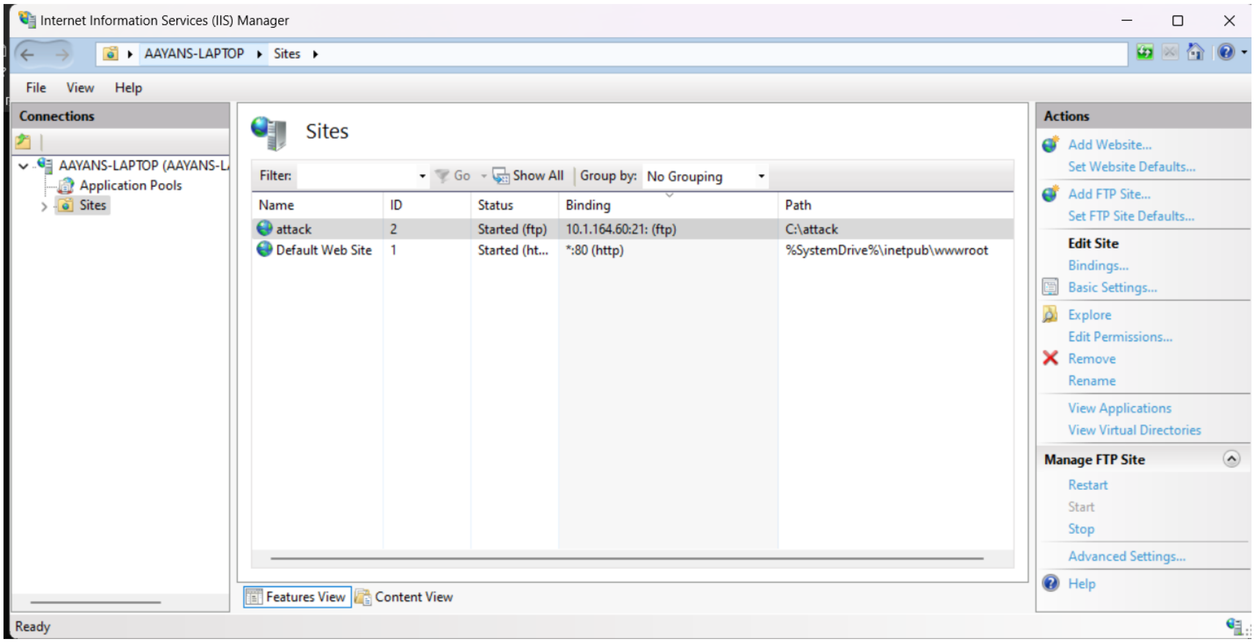
- net user <username> /active:yes



```
┌──(kali㉿kali)-[~]
└─$ ftp 10.1.164.60
Connected to 10.1.164.60.
220 Microsoft FTP Service
Name (10.1.164.60:kali): attack
331 Password required
Password:
530-User cannot log in.
 Win32 error:   The referenced account is currently locked out and may not be
 logged on to.
 Error details: An error occurred during the authentication process.
530 End
ftp: Login failed
ftp> bye
```

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.



Filter: | Group by: No Grouping

Name	ID	Status	Binding	Path
attack	2	Started (ftp)	10.1.164.60:21: (ftp)	C:\attack
Default Web Site	1	Started (ht...	*:80 (http)	%SystemDrive%\inetpub\wwwroot

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.
