FAWN Notes

Objective:

- The objective was to identify and exploit vulnerabilities within the **Fawn** machine on Hack The Box and retrieve a flag to submit.

Tools Used:

- **Nmap:** A powerful network scanning tool used for network discovery and security auditing.
- **FTP Client:** A command-line tool used to connect to FTP services and transfer files.

Methodology:

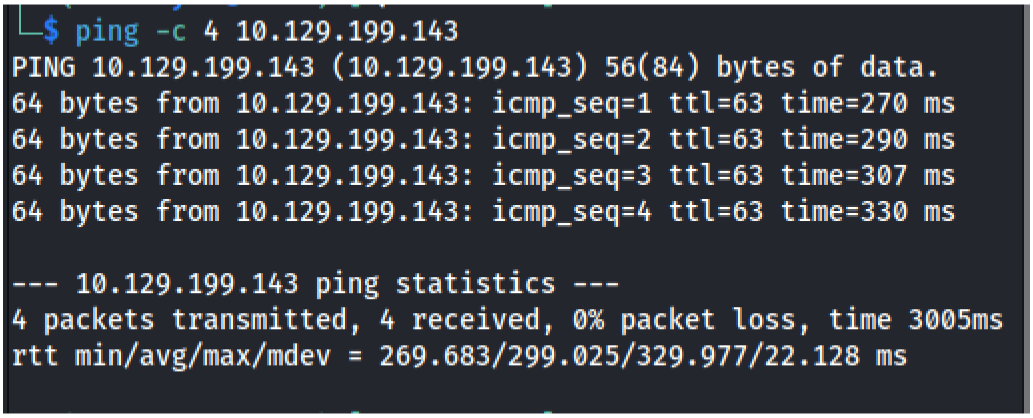
Information Gathering (Reconnaissance):

Ping the machine: I started off by pinging the machine to check if it was reachable by pinging its IP address:

```
ping -c 4 10.129.199.143
```

- **ping:** Sends ICMP echo request packets to the specified IP address to check if the host is reachable.
- **-c 4:** Sends 4 ping requests and then stops.
- **10.129.199.143:** The IP address of the target machine.

The target machine responded, indicating that it was up and reachable.



```
└─$ ping -c 4 10.129.199.143
PING 10.129.199.143 (10.129.199.143) 56(84) bytes of data.
64 bytes from 10.129.199.143: icmp_seq=1 ttl=63 time=270 ms
64 bytes from 10.129.199.143: icmp_seq=2 ttl=63 time=290 ms
64 bytes from 10.129.199.143: icmp_seq=3 ttl=63 time=307 ms
64 bytes from 10.129.199.143: icmp_seq=4 ttl=63 time=330 ms

--- 10.129.199.143 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 269.683/299.025/329.977/22.128 ms
```

Network Scanning:

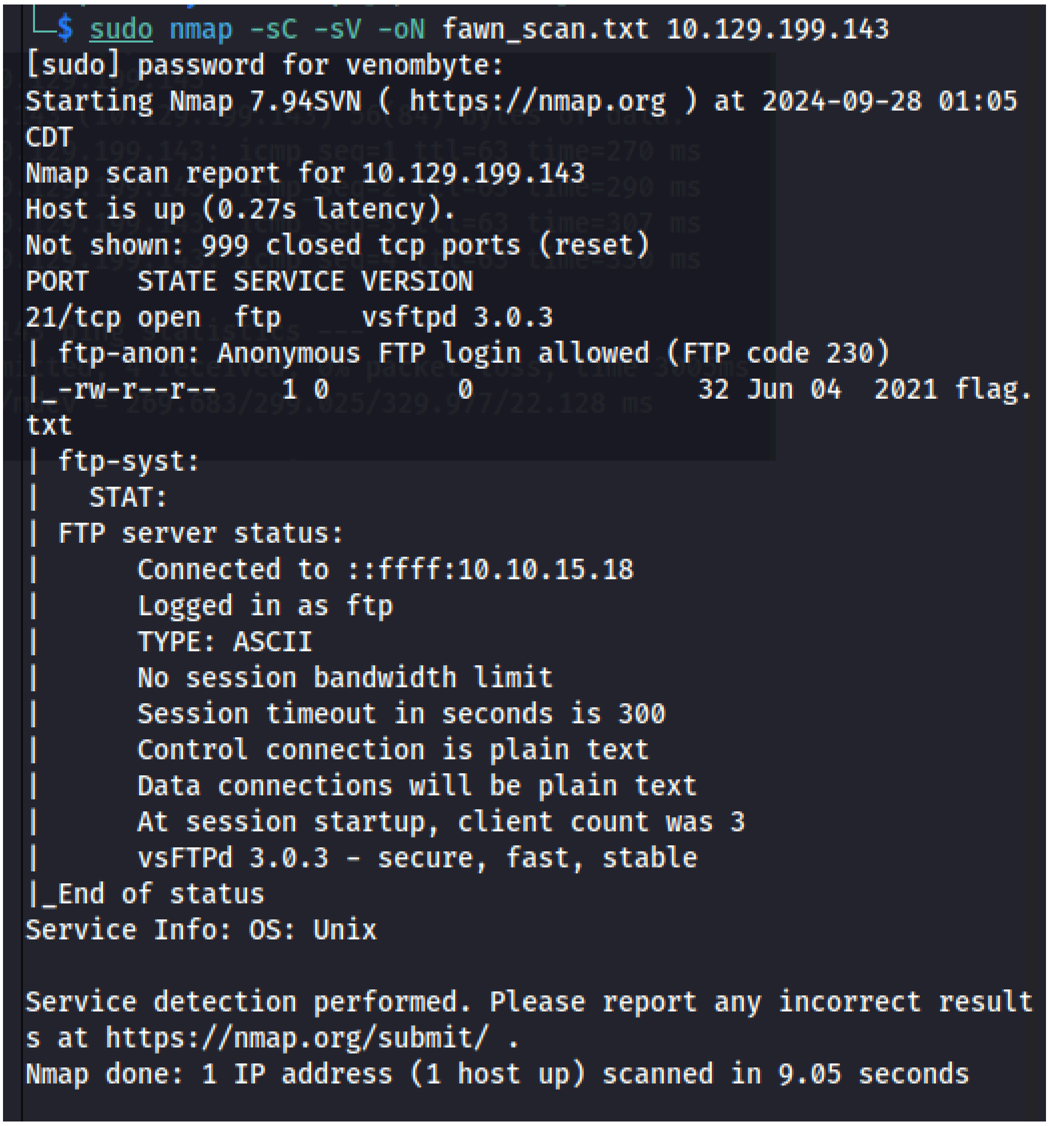
Nmap scan: After confirming that the machine was reachable, I proceeded with an Nmap scan to identify open ports and running services:

```
sudo nmap -sC -sV -oN fawn_scan.txt 10.129.199.143
```

- **sudo:** Runs the command with superuser privileges, which are often required for network operations.
- **nmap:** The command-line tool used for network discovery and security auditing.
- **-sC:** Runs default scripts to assist with service detection and vulnerability enumeration.
- **-sV:** Enables version detection, allowing Nmap to determine the version of services running on open ports.
- **-oN fawn_scan.txt:** Saves the scan results to a file named "fawn_scan.txt" for further analysis.
- **10.129.199.143:** The IP address of the target machine.

Nmap Results:

- The scan revealed that port **21** (FTP) was open and running **vsftpd 3.0.3**.
- The scan also indicated that **anonymous FTP login** was allowed.



```
└─$ sudo nmap -sC -sV -oN fawn_scan.txt 10.129.199.143
[sudo] password for venombyte:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-28 01:05
CDT
Nmap scan report for 10.129.199.143
Host is up (0.27s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 0        0              32 Jun 04  2021 flag.
txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.10.15.18
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
Service Info: OS: Unix

Service detection performed. Please report any incorrect result
s at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.05 seconds
```

Exploitation:

Connecting to FTP: After identifying that FTP was open and allowed anonymous login, I proceeded to connect to it using the following command:

`ftp 10.129.199.143`

- **ftp:** This command is used to connect to an FTP server.
- **10.129.199.143:** The IP address of the target machine.

Upon connecting, I logged in using **anonymous** as the username and no password (just pressing Enter). This granted me access to the FTP server.

Retrieving the Flag: Once connected, I listed the files in the directory using:

`ls`

This command revealed the existence of a file named **flag.txt**.

I then used the following command to read the contents of the flag:

`get flag.txt`

After downloading **flag.txt**, I opened the file to view its contents, which contained the flag.

Conclusion:

- The **Fawn** challenge made me aware about the risks associated with insecure configurations, such as allowing anonymous FTP logins. This configuration provided easy access to sensitive information on the system.

Recommendations for Securing FTP (Port 21):

To protect against unauthorized access to the FTP service, the following measures are recommended:

1. **Disable Anonymous Login:** The anonymous login feature should be disabled to prevent unauthorized users from accessing the FTP server. `anonymous_enable=NO`
2. **Use Secure Alternatives:** Replace FTP with **SFTP** (Secure FTP), which uses SSH to encrypt communication.
3. **Strong Authentication:** Enforce strong passwords for all accounts accessing the FTP server.
4. **Limit Access with Firewall Rules:** Use firewall rules to restrict access to the FTP server, allowing only trusted IP addresses. `sudo ufw allow from <trusted_IP> to`

`any port 21`

5. **Regular Audits:** Conduct regular audits of access logs and system configurations to ensure the server is secure.