最近比较衰，Redis数据库被好几次恶意攻击，导致里面的数据完全丢失。dump.rdb和appendonly.aof文件都不能恢复。一开始没有意识到数据库被攻击，后来几个数据库连续这样，才发现有问题。

被攻击后redis里面多了一个crackit表，打开一看发现对方通过入侵自己的redis数据库，然后利用dump文件获得服务器的root权限，企图使用自己的数据库挖矿（minerd，俗称挖比特币）。最近有大量的服务器被人入侵数据库，然后占用服务器的cpu挖比特币，所以对redis进行安全配置很有必要！很有必要！很有必要！千万不能疏忽！

大家一般使用Redis默认的.conf文件，默认端口是6379，没有认证密码，默认处理任何请求来源，这样会把自己的数据库完全暴露在外网环境中，十分危险。

**Redis安全配置**

对Redis进行安全配置主要从下面几个方面：

* 为redis添加**密码认证**

添加密码认证有临时的和长久的两种方法

1. 可以在redis运行时添加密码认证，在redis-cli连接redis后执行命令

CONFIG SET requirepass "myPassWord"

不过这种方式添加的密码认证只是临时的，在redis重启后密码就会失效

2.修改redis.conf文件，redis中默认注释掉了

requirepass mypassword

取消注释，然后将后面的字符串修改为自己的密码

密码越复杂越好吧

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 禁止一些**高危指令**

修改redis.conf文件，修改一些危险的命令。比如，在redis.conf中添加

rename-command CONFIG ""

rename-command FLUSHALL ""

rename-command FLUSHDB ""

rename-command EVAL ""

后面为空字符串相当于禁用了改命令，也可以根据自己需要将空字符串换成自己想要的命令重命名，比如：

rename-command CONFIG "CONFIG\_1"

但是需要注意一点的是，如果appendonly.aof文件中有执行过相应的修改数据命令，比如 FLUSHALL，那么命令重命名后可能导致数据恢复失败。

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 禁止外网访问Redis

修改redis.conf文件，里面有一条被注释的命令

bind 127.0.0.1

该条命令表示，redis只允许本地访问，如果注释掉改行或者将后面改为0.0.0.0，则表示任何IP均可访问。也可以根据自己需要，将后面的IP绑定为特定的IP访问，但是貌似只能绑定一个IP访问。比如：

bind 139.127.1.125

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 修改Redis默认端口

修改redis的端口，这一点很重要，使用默认的端口很危险，redis.conf中有一行为

port 6379

将其修改为自己指定的端口(可随意)，反正不使用默认的端口就好，比如

port 21122

* 1
* 2
* 3
* 4
* 5
* 6
* 7

**加密后Redis客户端访问**

加密后使用客户端访问可以直接使有两种方式

* 在访问时直接输入密码

redis-cli -h IP地址 -p 端口 -a 密码

* 1
* 先访问，然后再执行操作之前会提示需要密码

redis-cli -h IP地址 -p 端口

auth 密码

* 1
* 2
* 3
* 如果没有对CONFIG命令重命名，可以在使用密码登陆后使用命令获得密码

CONFIG GET requirepass

* 1

加密后java访问

如果是使用Jedis访问，那么可以参照代码：

Jedis redis=new Jedis("XXX.XXX.XXX.XXX",6379); //IP，端口

redis.auth("XXXXXX"); //密码

redis.ping();

如果是使用JedisPool访问，可以参照代码;

JedisPool Jedis pool = new JedisPool(config,REDIS\_IP,REDIS\_PORT,timeout,yourpassword); //timeout为读取超时，如果超过此事件没有读取结果就会报timeout异常

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

注意：redis配置修改后需要重启redis才会生效，重启方式：

redis-server redis.conf