**一：安全**

1：可信的环境

Redis以简洁为美。在安全层面Redis也没有做太多的工作。 Redis的安全设计是在“Redis运行在可信环境”这个前提下做出的。在生产环境运行时不能允许外界直接连接到 Redis 服务器上，而应该通过应用程序进行中转，运行在可信的环境中是保证Redis安全的最重要方法。

Redis的默认配置会接受来自任何地址发送来的请求，要更改这一设置，可以修改配置文件中的bind参数，如只允许若干IP连接Redis，可如下修改：

bind 192.168.1.100 10.0.0.1

bind 127.0.0.1

如果想更自由地设置访问规则需要通过防火墙来完成。

2：数据库密码

可以通过配置文件中的requirepass参数为Redis设置一个密码。例如：

requirepass TAFK**(**@**~!**ji**^**XALQ**(**sYh5xIwTn5D$s7JF

客户端每次连接到 Redis 时都需要发送密码，否则 Redis 会拒绝执行客户端发来的命令。发送密码需要使用auth命令，例如：

not connected**>** ping

**(**error**)** NOAUTH Authentication required**.**

127.0.0.1**:**6379**>** auth TAFK**(**@**~!**ji**^**XALQ**(**sYh5xIwTn5D$s7JF

OK

127.0.0.1**:**6379**>** ping

PONG

认证之后就可以执行任何命令了。由于Redis的性能极高，并且输入错误密码后Redis并不会进行主动延迟（考虑到Redis的单线程模型），所以攻击者可以通过穷举法破解Redis的密码，因此在设置时一定要选择复杂的密码。

3：命名命令

Redis 支持在配置文件中将命令重命名，比如将 flushall 命令重命名成一个比较复杂的名字，以保证只有自己的应用可以使用该命令。就像这样：

rename**-**command FLUSHALL oyfekmjvmwxq5a9c8usofuo369x0it2k

这种情况下，如果执行flushall命令，则会报错：

127.0.0.1**:**6379**>** flushall

**(**error**)** ERR unknown command 'flushall'

如果希望直接禁用某个命令，可以将命令重命名成空字符串：

rename**-**command FLUSHALL ””

注意，无论设置密码还是重命名命令，都需要保证配置文件的安全性，否则就没有任何意义了。

**二：通信协议**

Redis通信协议是Redis客户端与 Redis之间交流的语言，通信协议规定了命令和返回值的格式。了解Redis通信协议后不仅可以理解AOF文件的格式和主从复制时主数据库向从数据库发送的内容等，还可以开发自己的 Redis 客户端（不过由于几乎所有常用的语言都有相应的Redis客户端，需要使用通信协议直接和Redis打交道的机会不多）。

Redis支持两种通信协议，一种是二进制安全的统一请求协议(unified request protocol)，另一种是比较直观的，便于在 telnet 程序中输入的简单协议。这两种协议只是命令的格式有区别，命令返回值的格式是一样的。

1：简单协议

简单协议适合在telnet程序中和Redis 通信。简单协议的命令格式就是将命令和各个参数使用空格分隔开，如”exists foo”、”set foo bar”等。

由于 Redis 解析简单协议时只是简单地以空格分隔参数，所以无法输入二进制字符。比如：

root@localhost**:~**# telnet 127.0.0.1 6379

Trying 127.0.0.1...

Connected to 127.1.

Escape character is '^]'**.**

set foo bar

**+**OK

get foo

$3

bar

lpush plist 1 2 3

**:**3

lrange plist 0 **-**1

**\***3

$1

3

$1

2

$1

1

errcommand

**-**ERR unknown command 'errcommand'

在telnet程序中输入的5条命令恰好展示了Redis的5种返回值类型的格式，之前介绍过这5种返回值类型在redis-cli 中的展现形式，这些展现形式是经过redis-cli封装的，而上面的内容才是 Redis真正返回的格式：

a：错误回复，以”-“开头，在后面跟上错误信息，最后以”\r\n”结尾：

**-**ERR unknown command 'errcommand'

b：状态回复，以”+”开头，在后面跟上状态信息，最后以”\r\n”结尾：

**+**OK

c：整数回复，以”:”开头，在后面跟上数字，最后以”\r\n”结尾：

**:**3

d：字符串回复，以”$”开头，并在后面跟上字符串的长度和”\r\n”，接着是字符串的内容和”\r\n” ：

$3

bar

如果返回值是空结果nil，则会返回 “$-1”以和空字符串相区别。

e：多行字符串回复，以”\*”开头，后面跟上字符串回复的组数和”\r\n”，接着后面跟的就是字符串回复的具体内容了：

**\***3

$1

3

$1

2

$1

1

2：统一请求协议

统一请求协议，其命令格式和上面多行字符串回复的格式很类似，比如要发送命令

”set foobar hello”时，只需要按照上面介绍的格式发送即可。例子如下：

root@localhost**:~**# telnet 127.0.0.1 6379

Trying 127.0.0.1...

Connected to 127.1.

Escape character is '^]'**.**

**\***3

$3

set

$6

foobar

$5

hello

**+**OK

get foobar

$5

hello

**\***2

$3

get

$6

foobar

$5

hello

发送命令时指定了后面字符串的长度，所以命令的每个参数都可以包含二进制的字符。统一请求协议的返回值格式和简单协议一样，这里不再赘述。

Redis的AOF文件和主从复制时主数据库向从数据库发送的内容都使用了统一请求协议。如果要开发一个和Redis直接通信的客户端，推荐使用此协议。如果只是想通过telnet向Redis服务器发送命令则使用简单协议就可以了。

**三：管理工具**

1：redis-cli

作为 Redis 自带的命令行客户端，可以从任何安装有Redis的服务器中找到它，所以对于管理Redis而言，redis-cli是最简单实用的工具。

redis-cli可以执行大部分的Redis命令，下面介绍几个管理Redis时非常有用的命令：

a：慢查询日志功能

当一条命令执行时间超过限制时，Redis会将该命令的执行时间等信息加入慢查询日志(slow log)以供开发者查看。关于慢查询的配置选项有：

slowlog-log-slower-than 10000

slowlog-max-len 128

“slowlog-log-slower-than”参数指明记录慢查询的时间设置，单位是微秒（1000000微秒相当于1秒），负数表明将禁用慢查询日志功能，0表明将记录每个命令。注意这里的时间仅指命令执行时间，不包括IO时间（比如与客户端通信的事件）。

慢查询日志存储在内存中，可以通过“slowlog-max-len”参数来限制记录的条数。超过时，新命令将替换最早的命令。

使用”slowlog get”命令可获得当前的慢查询日志，为了验证该命令，将配置文件的“slowlog-log-slower-than”置为0。然后执行命令”set foo bar”和”get foo”，如：

127.0.0.1**:**6379**>** slowlog get

1**)** 1**)** **(**integer**)** 2

2**)** **(**integer**)** 1449296176

3**)** **(**integer**)** 15

4**)** 1**)** "get"

2**)** "foo"

2**)** 1**)** **(**integer**)** 1

2**)** **(**integer**)** 1449296173

3**)** **(**integer**)** 14

4**)** 1**)** "set"

2**)** "foo"

3**)** "bar"

每条日志都由以下4个部分组成：

该日志唯一ID；

该命令执行的Unix时间；

该命令的耗时时间，单位是微秒；

命令及其参数

b：命令监控功能

Redis提供了monitor命令来监控 Redis执行的所有命令，redis-cli支持这个命令，如在终端A的redis-cli中执行“monitor”，这时 Redis 执行的任何命令都会在终端A的redis-cli 中打印出来，如打开终端B的 redis-cli，执行”SET foo bar”命令，此时终端A的redis-cli中会输出如下内容：

127.0.0.1**:**6379**>** monitor

OK

1449296535.901203 **[**0 127.0.0.1**:**55287**]** "set" "foo" "bar"

注意，monitor命令非常影响Redis的性能，一个客户端使用monitor命令会降低Redis将近一半的负载能力。所以 monitor命令只适合用来调试和纠错。

2：phpRedisAdmin

当Redis 中的键较多时，使用redis-cli 管理数据并不是很方便，就如同管理 MySQL时有人喜欢使用phpMyAdmin一样，Redis同样有一个PHP开发的网页端管理工具phpRedisAdmin。

phpRedisAdmin支持以树形结构查看键列表，编辑键值，导入/导出数据库数据，查看数据库信息和查看键信息等功能。

3：Rdbtools

Rdbtools 是一个Redis 快照文件解析器。快照文件是二进制格式，不利于查看，可以使用Rdbtools来将其导出为JSON格式，分析 Redis 中每个键的占用空间情况等。Rdbtools 使用 Python 开发。