拓展 8：居安思危 —— 保护 Redis

本节我们来谈谈使用 Redis 需要注意的安全风险以及防范措施，避免数据泄露和丢失，避免所在主机权限被黑客窃取，以及避免人为操作失误。

指令安全

Redis 有一些非常危险的指令，这些指令会对 Redis 的稳定以及数据安全造成非常严重的影响。比如 keys 指令会导致 Redis 卡顿，flushdb 和 flushall 会让 Redis 的所有数据全部清空。如何避免人为操作失误导致这些灾难性的后果也是运维人员特别需要注意的风险点之一。

Redis 在配置文件中提供了 rename-command 指令用于将某些危险的指令修改成特别的名称，用来避免人为误操作。比如在配置文件的 security 块增加下面的内容:

```
rename-command keys abckeysabc
```

如果还想执行 keys 方法，那就不能直接敲 keys 命令了，而需要键入abckeysabc。
如果想完全封杀某条指令，可以将指令 rename 成空串，就无法通过任何字符串指令来执行这条指令了。

```
rename-command flushall ""
```

端口安全

Redis 默认会监听 *:6379，如果当前的服务器主机有外网地址，Redis 的服务将会直接暴露在公网上，任何一个初级黑客使用适当的工具对 IP 地址进行端口扫描就可以探测出来。

Redis 的服务地址一旦可以被外网直接访问，内部的数据就彻底丧失了安全性。高级一点的黑客们可以通过 Redis 执行 Lua 脚本拿到服务器权限，恶意的竞争对手们甚至会直接清空你的 Redis 数据库。

```
bind 10.100.20.13
```

所以，运维人员务必在 Redis 的配置文件中指定监听的 IP 地址，避免这样的惨剧发生。更进一步，还可以增加 Redis 的密码访问限制，客户端必须使用 auth 指令传入正确的密码才可以访问 Redis，这样即使地址暴露出去了，普通黑客也无法对 Redis 进行任何指令操作。

```
requirepass yoursecurepasswordherepleasse
```

密码控制也会影响到从库复制，从库必须在配置文件里使用 masterauth 指令配置相应的密码才可以进行复制操作。

```
masterauth yoursecurepasswordherepleasse
```

Lua 脚本安全

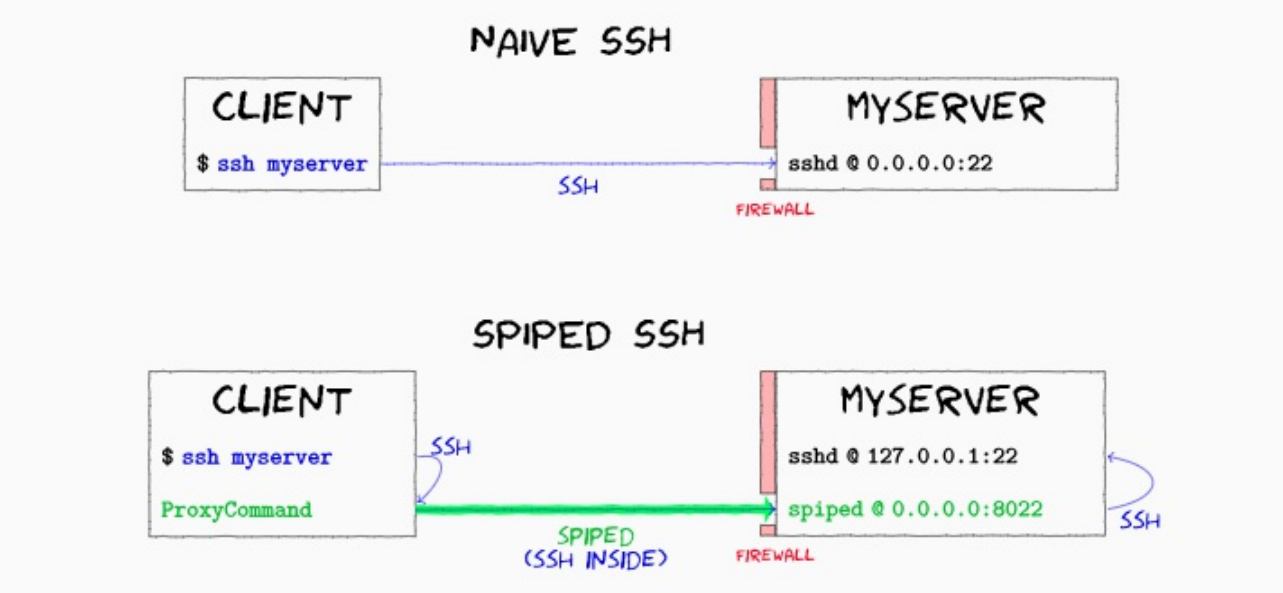
开发者必须禁止 Lua 脚本由用户输入的内容 (UGC) 生成，这可能会被黑客利用以植入恶意的攻击代码来得到 Redis 的主机权限。

同时，我们应该让 Redis 以普通用户的身份启动，这样即使存在恶意代码黑客也无法拿到 root 权限。

SSL 代理

Redis 并不支持 SSL 链接，意味着客户端和服务器之间交互的数据不应该直接暴露在公网上传输，否则会有被窃听的风险。如果必须要用在公网上，可以考虑使用 SSL 代理。

SSL 代理比较常见的有 ssh，不过 Redis 官方推荐使用 [spiped (http://www.tarsnap.com/spiped.html)](http://www.tarsnap.com/spiped.html) 工具，可能是因为 spiped 的功能相对比较单一，使用也比较简单，易于理解。下面这张图是使用 spiped 对 ssh 通道进行二次加密 (因为 ssh 通道也可能存在 bug)。



同样 SSL 代理也可以用在主从复制上，如果 Redis 主从实例需要跨机房复制，spiped 也可以派上用场。

小结

本节讲解了最基本的 Redis 安全防护思路，下一节我们来详细讲解 spiped 的原理和使用。