

[zz]“西厢计划”原理小解



从 farseerfc.wordpress.com 导入

好奇的想法，先存着，以后慢慢研究

原文：<http://blog.youxu.info/2010/03/14/west-chamber/>

待月西厢下，迎风户半开。隔墙花影动，疑是玉人来。

最近推上最流行的一个关键词是”西厢计划”，这个计划名字取得很浪漫，客户端叫做张生，对，就是西厢记里面那个翻墙去见崔莺莺小姐的张生；显然，服务器端必然叫做崔莺莺。客户端的张生是最重要的部件，可以不依赖于服务端工作。因为西厢计划的作者只是简要的介绍了一下原理，其他报道又语焉不详，我当时就觉得很好奇，花了昨天一个晚上详细读了一下源代码，终于知道怎么回事了，觉得原理非常漂亮，所以写篇文章介绍总结一下。

先说大方向。大家都知道，连接被重置的本质，是因为收到了破坏连接的一个 TCP Reset 包。以前剑桥大学有人实验过，客户端和服务器都

忽略 Reset, 则通信可以不受影响。但是这个方法其实只有理论价值, 因为绝大多数服务器都不可能忽略 Reset 的 (比如 Linux, 需要 root 权限配置 iptables, 而且这本身也把正常的 Reset 给忽略了)。只要服务器不忽略 Reset, 客户端再怎么弄都没用, 因为服务器会停止发送数据, Reset 这条连接。所以, 很多报道说西厢计划是忽略 Reset, 我从源代码来看应该不是这样。在我看来, 西厢计划是利用了墙的一个可能的弱点-墙只在连接发起的时候把一个 TCP 连接加入监听序列, 如果墙认为这个连接终止了, 就会从监听序列中去掉这条记录, 这样, 这条连接上后续的包就不会被监听。西厢计划就是让墙“认为”这个连接终止的一个绝妙的方法。只要墙认为这个连接两端都是死老虎, 墙就不会触发关键词检测, 其后所有的数据, 都不存在连接被重置的问题了。

如何让一个连接置之死地而后生, 就是西厢计划那帮黑客神奇的地方了。这也不是一日之功。首先, 这帮牛人发现, 墙的是一个入侵检测系统, 把含有关键字的包当成一种“入侵”来对待。采取这种设计有很多好处, 但缺点是入侵检测系统可能具有的问题, 墙都可能有。西厢计划主页上那篇著名的论文就是讲这些七七八八的漏洞的。可以说处理这些七七八八的漏洞是非常困难的, 迫使墙的设计者“拆东墙, 补西墙”。这样补来补去, 外表看起来好像很牛逼的墙, 其实有很多本质上无法简单修补的漏洞, 其中有一个致命的, 就是 TCP 连接状态的判定问题。出于入侵检测系统这种设计的局限, 墙没有, 也没办法准确判定一条 TCP 连接的状态, 而只是根据两边收到的数据来“推测”连接的状态。而所有的关键词检测功能, 都是基于“连接还活着”的这个推测的结果的。因为墙的规则是在连接发起的时候开始对这条连接的检测, 在连接终止的时候停止对这条连接的检测, 所以, 一旦对连接的状态推测错误, 把还活着的连接当成已经关闭的连接, 墙就会放弃对这条连接上随后所有的包的检测, 他们都会都透明的穿过墙的入侵检测。

上面只是想法, 具体到 TCP 协议实现这一层, 就要只迷惑墙, 还不能触及我要通信的服务器。最理想的情况下, 在任何有效通信之前, 就能让墙出现错误判断, 这些, 就需要对 TCP 协议有深刻理解了。西厢计划的那帮黑客, 居然真的去读 TCP 几百页的 RFC, 还居然就发现了方法

(这里我假设读者都知道 TCP 的三次握手过程和序列号每次加一的规则)。我们都知道, 三次握手的时候, 在收到服务器的 SYN/ACK 的时候, 客户端如果发送 ACK 并且序列号+1 就算建立连接了, 但是客户端如果发送一个序列号没 +1 的 FIN (表示连接终止, 但是服务器知道, 这时候连接还没建立呢, FIN 这个包状态是错的, 加上序列号也是错的, 服务

器自己一判断，就知道这个包是坏包，按照标准协议，服务器随手丢弃了这个包），但这个包，过墙的时候，在墙看来，是表示连接终止的（墙是 made in china, 是比较山寨的，不维护连接状态，并且，墙并没有记下刚才服务器出去的 SYN/ACK 的序列号，所以墙不知道序列号错了）。所以，墙很高兴的理解为连接终止，舒了一口气去重置其他连接了，而这个连接，就成了僵尸，墙不管你客户端了，而这时候，好戏才刚刚开始。

事实上，墙是双向检测的（或者说对每个包都检测的），因此，对服务器和客户端实现相同的对待方法，所以，墙不管客户端还不行，假如服务端有关键词传给客户端，墙还是有可能要发飙的（这里说有可能，因为我也不知道）。所以，最好的办法就是，让服务端也给墙一个终止连接的标志就好了。可是这个说起来简单，做起来难，怎么能让不受自己控制的服务器发一个自己想要的包呢？西厢计划的那帮黑客，再次去读几百页的 RFC, 令人惊讶的发现，他们居然在 RFC 上发现了一个可以用的特性。我们上面说了，三次握手的时候，在收到 SYN/ACK 后，客户端要给服务器发送一个序列号+1 的 ACK, 可是，假如我不+1呢，直接发 ACK 包给服务器。墙已经认为你客户端是死老虎了，不理你了，不知道你搞什么飞机，让这个 ACK 过了。可是服务器一看，不对啊，你给我的不是我期待的那个序列号，RFC 上说了，TCP 包如果序列号错了的话，就回复一个 Reset. 所以，服务器就回复了一个 Reset。这个 Reset 过墙的时候，墙一看乐了，服务器也终止连接了，好吧，两边都是死老虎了，我就不监听这条连接了。而至于客户端，这个服务器过来的 Reset 非常好识别，忽略就是。随后，客户端开始正确的发送 ACK, 至此，三次握手成功，真正的好戏开始，而墙则认为客户端和服务端都是死老虎，直接放过。所以，张生就这样透明的过了墙。至于过墙以后所有的事情，《西厢记》里面都有记载，各位读者自行买书学习。

现在的西厢计划客户端，即“张生”模块的防连接重置的原理就是这样，服务器端，即莺莺模块的实现也是类似的。防 DNS 那个，不懂 DNS 协议，所以看不懂。我猜想，因为开发人员都是黑客，所以自然喜欢用最经得起折腾和高度定制的 Linux 开发。现在看西厢计划的实现，因为依赖于 Linux 内核模块 netfilter, 在 Linux 上如鱼得水，但往其他平台的移植可能是个亟待解决的问题。我觉得，在其他平台上，可以通过 libpcap 和 libnet，在用户态实现相同的功能，就是有点麻烦而已，有兴趣的懂网络的可以照西厢计划原理，在家自行做出此功能；当然，全中国人民都用 Linux 最好：)

PS 1: 据说是西厢计划一个作者画的原理图：<http://img.ly/Dli>

PS 2: 我对 TCP 的理解仅限于课本，如果上面的对技术的理解有错，请大家指出。

PS 3: 有些漏洞，可能是设计上本质缺陷，不是那么容易修复的。

PS 4: 除了最后一个图，本文没有其他相关链接，如需相关资料，自行 Google。