Freebsd+ipfw2+Apache 有时会有一些问题：当系统运行一段时间后,比如几天或者几周后,用命令 netstat -an 会看到许多处于 fin_wait_2 状态的死连接,这些连接随着时间增长会越来越多,而且似乎无法消除,无论用 ipfw 进行防火墙规则的重新设定（比如 ipfw flush 一下),或者用 ifconfig 对网卡进行 down 或者 up 都无法做到消除这些连接,(不知道有没有清除 tcp 状态表的工具),所以只有重新启 动系统才行.而如果系统不设置动态防火墙 rule,则不会有此现象.

我在 2005 年的时候在 ivan 的一台 FreeBSD 64 位机器上碰到过这个情况,当时是非常头大的,虽然不影响使用,但总是感觉不爽， 从 google 上找到一些东西,但都没有说清楚,其中有一篇文章,最后找到 freebsd ipfw 的专家 **Luigi Rizzo** 写在一个 maillist 上面的留言,我看了这个留言后才对这个问题有稍微清楚的认识:

By **Luigi Rizzo**: http://lists.freebsd.org/pipermail/freebsd-ipfw/2003-May/000206.html

i imagine the following happens:
+ the client does not properly close the connection;
+ when a keepalive is sent (every 5 minutes), the the server's TCP responds (thus refreshing the rule), and the TCP timeout
is reset so it stays in the FIN_WAIT[2] state for another cycle, whereas the client does not bother to send back a RST (which would cause the timeout for the dynamic rule go down to very low values).

This would explain why the phenomenon is relatively rare (500 entries in 5 days).

Maybe i should change the logic in the dynamic rules so that further keepalives are not sent unless a reply has been received from both sides.

当时在那台机器上产生这些连接的基本上都是 Apache 服务,它作为一个 server 端,能够进入 fin_wait_2 状态,只能是因为它发出了 active close,即主动结束 tcp 连接. 这在网页传输过程中应该很正常的,因为数据传完了,它必须要告诉客户端:我传完了,我要结束连接,它先向 client 发个 fin,然后进入 fin_wait_1 状态，等待 client 发来 ack。一旦接受到 ack,马上进入 fin_wait_2 状态， 等待 client 也发个 fin 过来， 然后 ack,然后进入 time_wait 状态等待 2 个 MSL， 然后消失。这种是正常的结 束。但是假设 server 进入 fin_wait_2 状态后 client 端没有进行 graceful close,即它根本不向服务端发 fin,或者 client 发的 fin 被某种通道上的东西挡住了(或者丢失了)，那么 server 端就会以 fin_wait_2 状态等待下去， 等待的时间长短由 tcp 的一些参数 决定， 不同平台不同,超过这个时间连接就会自动消亡,不会一直赖着不走的. 但是当系统

里面加载了 ipfw 动态规则就不一样了 ipfw 默认对创建 dynamic rules(动态规则)会发 keepalive packets，即它会保持这个连接!!在 ipfw 的 man 说明里面有这么一段:

net.inet.ip.fw.dyn_keepalive: 1

Enables generation of keepalive packets for **keep-state** rules on TCP sessions. A keepalive is generated to both sides of the connection every 5 seconds for the last 20 seconds of the lifetime of the rule.

也就是说,ipfw 默认在动态规则的生存期(lifetime)的最后 20 秒里面,每隔 5 秒(?)会给 Client,Server 分别发送 keepalive 信号.

我们来看看 FreeBSD 系统的默认 lifetime,摘自 ipfw's man:

net.inet.ip.fw.dyn_ack_lifetime: 300 //????? Why this? 注 1
net.inet.ip.fw.dyn_syn_lifetime: 20
net.inet.ip.fw.dyn_fin_lifetime: 1 // ?? Not this ??
net.inet.ip.fw.dyn_rst_lifetime: 1
net.inet.ip.fw.dyn_udp_lifetime: 5
net.inet.ip.fw.dyn_short_lifetime: 30 //but not always the same on all system

这个就是问题的根本了,对于 fin_wait_2,根据参数推测，会等 net.inet.ip.fw.dyn_ack_lifetime: 300 秒左右(也就是上文说的 5minutes)<-见注 1 而它在最后 20 秒又自动给 client,server 发 keepalive 信号，server 的 Tcp 回应了,timeout 被重置，又会等 300 秒，而 client 端对这个"无故发来"的 keepalive 信号不理会,因为它认为上次连接已经结束,它没有义务对这个信号回应(极有可能是装了防火墙 了)，哪怕它发个 RST 也好,这个连接也能结束,但是它没有发，于是 Server 端就在那里一直傻等.....就像谈恋爱分手后一方对另一方的感觉，可见,明确通知对方"我不爱你了" 有多么重要阿!!!

(注 1:这个到底要等多少秒,我没有查到结果,可能要看源代码才能看到是哪个rule的lifetime在起作用. 根据apache网站的一篇文章:http://httpd.apache.org/docs/1.3/misc/fin_wait_2.html,里面说到:在标准的RFC里面是**没有**FIN_WAIT_2 的timeout设定的,所以造成某些client如果不发fin的话,一些老的OS的server端就永远不退出,因为它没有这个timeout概念. FreeBSD从 2.0 后已经加了这个fin_wait_2 timeout,所以应该没有事,问题就是ipfw2 搀和进来了,它似乎overwrite了系统的timeout(或者小于系统的),所以造成上面的情况,上面我引用的Rizzo的帖子后面还有一贴:http://lists.freebsd.org/pipermail/freebsd-ipfw/2003-May/000207.html ,这是**Gregory Neil Shapiro**对Rizzo 的话的评论,其中有一个疑问和我的一样:"But wouldn't a dyn_fin_lifetime of 1 mean it wouldn't reach 5 minutes?",看来ipfw2 在这里采用的并不是字面上的意思,它lifetime是指上一个包的类型,上一个包过来后这条动态规则还要存在相应的延时，具体到这里的

dyn_ack_lifetime指的是上一次client发来的用于结束fin_wait_1 状态的是 ACK 包,而不是将要接收的Fin包,所以这个规则离消亡又要等一个 300 秒 (dyn_ack_lifetime)...)

The solution:
那么解决问题的方案之一就是不要发 keepalive

Found TOO many fin_wait_2 connections ? Solution is :
ipfw diable dyn_keepalive

或者(or:)

sysctl net.inet.ip.fw.dyn_keepalive=0

注意,要等一会,大概在 5 分钟以后,你会慢慢地看到 netstat -an |grep FIN_WAIT 消下去.

And wait for 5min or more (depending on your settings of *_lifetime), the [netstat -an | grep FIN_WAIT] output result would slowly goes down...
(请原谅我一段英文,一段中文的,老毛病了,写 regshot 多语言版的时候养成的,便于交流)

Free BSD 专家:Luigi Rizzo 的最后一段话是最好的解决办法,我一直没有用新版本的 freebsd 和 ipfw,不知道现在改进没有.

by regshot 2005