嵌入式防火墙上实现防御 SYN Flooding 攻击

杨 淼，高 燕
（解放军信息工程大学 通信工程系，河南 郑州 450002）

摘 要：论述了一种复合型嵌入式防火墙的实现结构，分析了 SYN Flooding 攻击原理及其多种防御措施，在 FPGA 上运用 CUSUM 算法实现了对 SYN Flooding 攻击的检测。

关键词：SYN Flooding SYN Cookies CUSUM 算法

在目前复杂的网络环境下，安全成为关注焦点，而防火墙是维护内部网络安全的首选产品。为获得更高安全性，常常把基于包过滤的方法与基于应用代理的方法结合起来，形成复合型防火墙。不仅如此，网络攻击行为的泛滥也使得抗 DDOS 攻击的功能被集成到防火墙中。SYN Flooding 是一种使用频繁的 DDOS 攻击手段，发起这种攻击并不复杂，有多种可选择的攻击软件，甚至利用 Raw Socket 编写一个攻击程序也非难事[1]，而且它的攻击是利用 TCP/IP 协议设计上的安全缺陷，至今还没有彻底的解决方案。一种可行的解决方式是综合利用各种手段抑制和减少攻击带来的损失，提高防火墙的抗攻击能力。本文介绍一种在嵌入式防火墙上实现防御 SYN Flooding 攻击的方案。

1 嵌入式防火墙平台

本文所述防火墙主要是为中小型单位设计的，这些单位的内部网络规模通常较小，网络应用并不复杂，防火墙的设计力求达到管理简单、相对功能齐全、能防御常见攻击类型。硬件平台采用 Intel IXP425 开发板，板上集成有 FPGA 芯片，开发板支持一个 WAN 网络接口和九个 LAN 网络接口，另外还有 USB 接口、串口、20 针 JTAG 接口等。

所有流量首先经 MAC 层处理，然后被 FPGA 上的检测引擎捕捉，提取参数，交给网络处理器（NP）处理。防火墙的功能结构图如图 1 所示。

在任何情况下，内外网都不能直接建立连接。在 IP 层，网络流量进入处理器之后，首先经状态追踪和 DNAT 处理，然后根据目的地址进行路由输入，把数据包交给输入过滤器或转发过滤器，按事先定义的策略选择包并决定是否交给上层或转发；与此同时，上层所产生的数据包交给输出链进行状态追踪和过滤，然后进行 SNAT 处理。在应用层，启动代理服务器机制，内网发向外网的请求在 IP 验证和访问权限验证以后被代理。

2 SYN Flooding 攻击原理分析

在正常的 TCP 连接请求中，如果服务器（Server）收到客户端（Client）发来的 SYN 包，就不加验证地回应 SYN/ACK 包，同时也为该连接分配系统资源以维护连接状态。在服务器 SYN-RECVD 状态超时以前，如果收到客户回应的 ACK 包，则服务器从该状态进入 ESTABLISHED 状态，三次握手完成。在这个连接过程中，若客户端以伪造的 IP 地址发出 SYN 包，服务器就不可能收到回应的 SYN/ACK 包，它也就无需为连接负担开销。而服务器端所分配的空间，在客户端刻意不给回应 ACK 包的情况下，直到服务器 SYN-RECVD 状态超时，才能释放掉。正是这种资源分配上的不对称，使得攻击者可以发出大量伪造的 IP 地址的数据包以达到两个目的：一是服务器的系统资源耗尽，二是客户的真正连接请求报文被淹没，服务器拒绝建立连接。

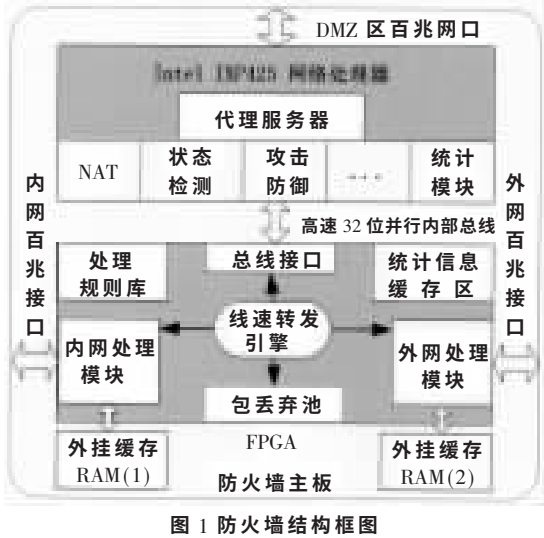

图 1 防火墙结构框图

3 在 NP 上实现的防御措施

根据以上分析首先可以得出结论：修改 TCP/IP 协议栈中的参数值，减少 SYN-RECVD 状态超时时间和增大系统 Backlog 大小，可以增强防火墙的抗攻击能力。但是超时时间设置过短，会使正常连接也被拒绝连接；扩大缓存则需要硬件资源支持，不能无限增加。缓存所需

容量可依据 $L=S\times\left[\frac{B}{64\times8}\right]$ 来估计，L 是队列长度，S 是半打开连接超时时间，B 是连接到防火墙的带宽。本文选择的队列长度小于这个估值。

从防火墙的工作流程来看，所有外网接口过来的连接请求，首先要根据防火墙中的访问控制规则过滤掉一部分攻击报文，然后在防火墙中启动应用代理。任何时刻内外网之间都不能建立直接通信连接，必须由应用代理进行处理、转发内网和外网通信，外网发过来的连接请求包只能到达这里，整个攻击集中在防火墙。这样做可使服务器不加改动，同时也能保护整个内网。

在 NP 上对防御 SYN Flooding 攻击起关键作用的是 SYN Cookies。SYN Cookies 技术最先由 D. J. Bernstein and Eric Schenk 提出并运用[2]，现在已经发展成熟。它的思想是在连接请求到来的时候，产生一个 Cookie 作为己方初始连接序号（ISN）回应给连接发起方，并不为这个未完成队列分配资源，（实际上是把连接状态存于密码即 Cookie 当中）。第三次握手，客户返回这个 ISN 并加上 1，被动连接方要验证 Cookie 以决定该连接可否接受。通过对 TCP/IP 协议栈的修改，这种方式改变了普通 TCP 实现中请求方和被请求方资源分配不公的状况，使攻击失去发起攻击的前提。从连接过程看，它允许防火墙在缓存队列填满的情况下，仍然能接受新的连接，提高了对攻击的防护性能。

在 Linux 内核中，由 random.c 文件中的 secure_tcp_syn_cookie 函数计算 Cookie，而 Cookie 的验证则在 check_tcp_syn_cookie 函数中进行。产生 cookies 的算法如公式（1）：

$$\begin{aligned}&A=hash_{32-61}(S_{addr}|S_{port}|D_{addr}|D_{port}|K_1)\\&B=hash_{32-61}(S_{addr}|S_{port}|D_{addr}|D_{port}|counter|K_2)\\&cookie=A+ISN_c+(counter\times2^{24})+(B+data)\bmod 2^{24}\end{aligned}\quad(1)$$

收到 ACK 包时验证算法如公式（2）：

$$\begin{aligned}&counter_{cookie}=(cookie-A-ISN_c)/2^{24}\\&Dcounter=counter_{current}-counter_{cookie}\\&\text{data}=cookie-A-ISN_c)\bmod2^{24}-\text{B}\bmod 2^{24}\end{aligned}\quad(2)$$

在 SYN Cookies 机制中，MSS（Maximum Segment Size）只能取八种预定义的值，data 值是用于存储 3bit 的 MSS 值。密钥 K_1、K_2 使用内核的随机数字发生器产生，这两个值在初始化后作为常量保存在内核的存储器中，直到内核重新引导。这种对 TCP 协议栈的改进，无需客户端参与，且对通信过程无影响。

防火墙收到客户真正返回的 ACK 包之后，可以反向计算出 counter 和 data 的值。理论上，计算结果和计算 cookie 时所使用的值相符即可证明该连接合法，但由于初次使用的值并未被保存，在实际操作中，TCP 服务器要首先计算出当前分钟值，推算出差值 Δcounter，差值在 4 分钟以内可以初步认定所返回的 ACK 包有效。然后，看计算出的 data 值是否合法。两个判定条件都满足，该连接被接受，加入连接队列。

SYN Cookies 机制的弊病是除 MSS 之外的 TCP 选项信息均被简单忽略，这些选项信息包括为提高 TCP 连接性能而设的窗口扩大因子和时间戳；同时它也不支持 TCP 扩展 T/TCP 协议。在实际应用中，采取只有当未完成连接队列已满时才启动 SYN Cookies 的策略以图减少其负面作用。

4 在 FPGA 上实现的防御措施

以上防御手段可以满足在一定攻击强度下防火墙仍能正常工作的目的。但是随攻击强度的增加，需要其他手段配合以挫败攻击。由图 1 可以看出，外网过来的流量首先进入 FPGA，所以这里是统计检测异常流量的理想地点。在 FPGA 上实现一个精确检测算法，使之能够在攻击达到一定强度时，由系统发出警报，提示管理员采取限制外网访问流量等手段来保护网络。实现框图如图 2 所示。

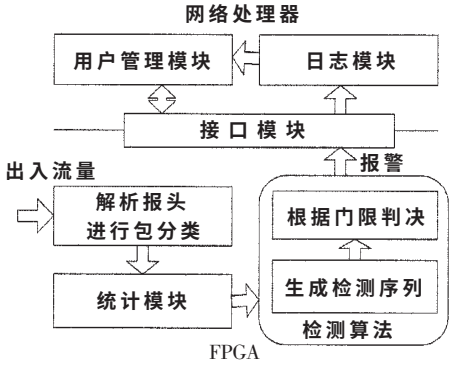

图 2 检测算法实现流程图

4.1 检测算法要求

期望得到这样一种检测算法：认为 SYN 包的到达速率在某个跳变点以下不必警惕，只关注超出这个跳变点以后的网络行为；如果在一段时间内，攻击包到达速率持续超出跳变点，并超过检测门限，算法在检测时延约束下能检测到真实攻击的发生；并且希望那些只在短期内突发偶然超出跳变点的行为不被认为是攻击，以求降低虚警率；在攻击结束以后，检测值能迅速降低至正常水平并不对后续的检测结果造成影响。从系统资源的角度考虑，检测算法若以保存连接状态为前提，那么在强攻击到来时需要耗费大量资源，且算法本身很脆弱；使用 FPGA 硬件进行检测，其实现优势在于实施线速处理，但 FPGA 设计中存储资源又是稀缺的，因此本文设计的实现结构采用无状态的检测算法更为适合。

基于以上考虑，本文选择工控领域改变点检测常用的非参数 CUSUM（Cumulative Sum Algorithm）算法。

4.2 检测特征量的选取

确定所要采用的算法以后，还有一个关键问题就是检测什么？统计模型中常用的检测量包括：检查事件的数量、到达间隔、资源消耗情况等。通过考察正常 TCP 连接过程中的行为，选取能够恰当刻画攻击行为的特

征。在正常通信过程中,SYN 包和 FIN 包定义了 TCP 连接的开始和结束,它们强正相关[3],SYN 包的出现会引起 FIN 包的发送,几乎是一一对应的。当攻击发生时会出现大量的 SYN 请求包,根据数量异常就可以发现攻击开始。在一个检测周期内,SYN 和 FIN 对之间也会存在差值,原因如下:存在连接持续时间大于检测周期的会话和 RST 包的出现。复位报文 RST 包用于出现错误的场合[4],包括连接出错后以 RST 包中止一个连接,这时 RST 包的出现和 SYN 包相关联,它可以由连接的任意一方发出,收到的一方则释放掉资源并不做出回应。为准确描述攻击特征,这里用 SYN/FIN+RST 对之间的差值作为检测量。

检测是在外网接口进行,主要监视出入两个方向的流量。这里并不区分 SYN 包和 SYN/ACK 包,一概作为 SYN 包进行统计,连接双方都会发出的 FIN 包也都相应地计算在内。这样就产生一个问题:如果以 RST 包结束一个连接,RST 包只出现一次,而且 RST 包并非都和 SYN 包相关,如发往错误端口的数据包。在实际观察中发现,RST 包和 SYN 包的正相关性也很强,如在 http 业务中,在长时间没有数据交换的情况下,常有以 RST 包释放连接的情况,这里 RST 包的出现和 SYN 包相关。但是如果认为所有 RST 都和 SYN 相关,算法就会不敏感,如果认为所有 RST 都和 SYN 不相关联,算法必然虚警率高,加上考虑长期存在的连接对连接请求包与连接结束包之间差值的影响,取经验门限值为 0.75[3],即认为只有 0.75 的 RST 包和SYN 包关联。差值定义如公式(3)。

$$\Delta n=SYN_n+SYN/ACK_n-FIN_n-1.5RST_n,\ n=0,1,2\cdots \tag{3}$$

式中,n 是离散时间序列。

4.3 检测算法

网络流量有很强的突发性和不稳定性,在不同的工作日,以及一天内的不同时刻,流量差别很大。为去除网络规模和抽样时刻对检测序列的影响,对差值做如下标准化处理:

$$X_n=\frac{\Delta n}{F_n}$$

$$F_n=\lambda F_{n-1}+(1-\lambda)F_n,\ 0<\lambda<1 \tag{4}$$

式中,F_n 是抽样区间内 FIN 包和 RST 包(取 1.5 倍)个数和的递归运算。

随机过程经过标准化处理后去掉了相关性,是平稳随机过程。正常情况下随机变量 X_n 的均值 $E(X_n)=c$,$c<<1$,接近 0。而选取因子 a 则按式(5)改造变量 X_n,得到一个能满足 CUSUM 算法的新变量 Y_n,它的均值在未发生攻击时为负,而异常发生时,均值迅速变成正值。

$$Y_n=X_n-a,\ a>0 \tag{5}$$

a 就是正常情况下随机变量 X_n 均值所能达到的上界,经改造之后,Y_n 均值在正常情况下为负。z_n 是实际检验统计量,代表 Y_n 的累积正值(参见式(6))。引入因子 a 之后,正常状态下检验统计量 z_n 常常归零,不会随时间累加。

$$z_n=(z_{n-1}+Y_n)^+ \tag{6}$$

检测中还需要一个判决门限,认为超出门限的行为是异常的。判决函数 $d_n(z_n)$ 定义如下:

$$d_N(z_n)=\begin{cases}0 & (z_n\le N)\\ 1 & (z_n>N)\end{cases} \tag{7}$$

式中,N 是判决门限

累积正值 z_n 没有超出门限,即 $z_n\le N$,认为情况仍属正常($d_N(z_n)=0$);一旦超出门限,即 $z_n>N$,判定攻击发生($d_N(z_n)=1$),此时根据事先定义的信号,产生报警信息,通知系统管理员。

4.4 算法性能研究及应用参数选取

检测需有两个基本性能指标:(1) 无虚警持续时间(连续两次发生虚警的时间间隔)。(2)检测时间(攻击发生后的检测延迟)。先讨论无虚警持续时间,当检测门限增大并趋于无穷时有:

$$P_\infty\{d_N(n)=1\}=\alpha\exp(-\beta N) \tag{8}$$

式中,P_∞ 是概率;α 和 β 只起次要作用,实际上可以忽略,则由(8)式可看出连续两次虚警之间的间隔时间随参数 N 呈指数增长。

为研究检测时间,定义:

$$\tau_N=\tau(d_N)=\inf\{n:d_N(\cdot)=1\}$$

$$\rho_N=\frac{(\tau_N-m)^+}{N} \tag{9}$$

式中,τ_N 是检测出攻击的时刻,m 是序列实际发生变化的时刻,也就是攻击真正开始的时刻。$(\tau_N-m)^+$是检测延迟,ρ_N 是标准化的检测延迟。假定发生攻击的情况下,Y_n 均值至少为 h(下界),则 h 和 ρ_N 有下述关系:

$$\rho_N\to\gamma=\frac{1}{h-|c-a|},\ h>a \tag{10}$$

式中,$h-|c-a|$是攻击开始后 Y_n 的均值。由上式计算出的是最大归一化检测时延。由于 h 是定义的最小值,(10)式就是检测延迟的保守估计,即最大检测延迟。在检测时总希望在一定检测延迟约束下,保证无虚警时间越长越好,(8)、(9)两式有助于为达到此目的而合理选取参数。

算法中有两个可变参数在应用时需要明确,即正常情况下的上界 a 和攻击时的下界 h。由上面的讨论可以推出,为满足一定的检测延迟约束和无虚警时间,N 的大小要选择合适,而 h 的大小随 N 而变化。参考文献[3]中的应用,取 $h=2a$。

在防火墙与外网接口为百兆带宽且所有流入流量必须经过此接口的实际情况下,每秒钟最多能到达 20 000 个最小尺寸攻击包,和正常应用下的差值存在非常大的差异。在商业领域的 SYN 包攻击的实验中,以最小洪泛攻击速率覆盖一个不受保护的服务器需要 500 个洪泛包/每秒,配以专防攻击的防火墙,需要 14 000 个包/每秒[4]。这样,a 的选取对算法性能影响并不敏感。由于防

(下转第 134 页)

(上接第 131 页)

火墙本身有抗攻击机制可以对抗一定攻击强度下的攻击,无需算法过于敏感,选择时有很大余地,这里选 $a=0.5$,则 $h=1$。设 $c=0$,允许的最大检测延迟为两个统计周期,即取值为 2。由公式(9)、(10)可以推得 $N=1$。

应用时还需要明确统计周期。流量测量显示,一般 TCP 连接持续 12~19 秒,这里取一个统计周期为 30 秒,为使统计量同属一个周期,之间相关性更强,则把 FIN 包和 RST 包的统计时刻推后 15 秒。

本算法检验的是洪泛攻击的累加效果,所以无论攻击方式是突发还是持续增加,其检测性能并无区别。经检验,运用本算法完全可以达到防火墙的设计指标。

SYN Flooding 攻击是目前很流行的攻击方式,易于实施且难于防备。有很多防范方法,但都不能完全奏效,只有结合各种方法的长处,避开其短处,才是实际解决之道。本文针对 SYN Flooding 攻击,采用嵌入式复合型防火墙平台,在网络处理器上利用安全嵌入式 Linux 内核并改进 TCP 协议栈进行实现,启用 SYN Cookies,在外网端口上使用 FPGA 硬件逻辑进行线速无状态 CUSUM 检测,整体实现正是基于这种综合防御的思想。

参考文献

[1] STEVENS W R. TCP/IP 详解:卷 1[M].范建华,译.北京:机械工业出版社,2004.

[2] Syn cookies mailing list,syncookies-archive@koobera.math.uic.edu[EB/OL].http://cr.yp.to/syncookies/archive.

[3] WANG Haining, ZHANG Danlu, SHIN K G. Detecting SYN flooding attacks[C]. Proceedings of IEEE INFOCOM' 2002. New York:[s.n.],2002.

[4] DARMOHRAY T, OLIVER R. Hot spares for DoS attacks [M]. Login, 2000.

(收稿日期:2006-09-21)