基于 Linux 平台的新的 SYN Flood 防御模型研究

刘 云

(贵阳学院数学与信息科学学院 贵阳 550005)

摘 要 SYN Flood是一种典型的拒绝服务攻击技术,它利用TCP协议的安全漏洞危害网络,目前还没有很好的办法彻底解决SYN Flood攻击问题。分析了3种现有的SYN Flood防御模型:SYN Cookie、SYN Gateway和SYN Proxy,提出了增强的SYN Proxy防御模型,研究了其相关的防御算法,并基于Linux平台进行了实现,最后对防御模型进行了测试。测试结果表明,增强的SYN Proxy模型能抵御高强度的SYN Flood攻击,较之现有的模型有更好的优越性。

关键词 SYN Flood攻击,SYN Cookie,增强的SYN Proxy模型,Linux平台,握手信息

中图法分类号 TP393.08 **文献标识码** A

Research of New SYN Flood Defense Model Based on Linux

LIU Yun

(College of Mathematics and Information Science, Guiyang University, Guiyang 550005, China)

Abstract The SYN Flood is a typical denial of service attack technology and endangers the network using the security vulnerabilities of the TCP protocol. There is no good way to completely solve it at present. This paper analyzed the three existing SYN Flood defense model: the SYN Cookie, the SYN Gateway, the SYN Proxy, and put forward the enhanced SYN Flood defense model, and researched the related algorithm, and implemented the model based on linux, and tested the defense model last. The result of the test shows that the enhanced SYN Proxy model can resist the high intensity SYN Flood attack and be better superiority than the existing model.

Keywords SYN Flood attack, SYN Cookie, Enhanced SYN Proxy model, Linux platform, Handshake information

TCP协议创建连接的三次握手的过程是:首先客户端发送SYN报文到服务端请求连接,然后服务端回复SYN ACK报文进行应答,最后客户端发送ACK报文到服务端确认建立连接[1]。SYN Flood正是利用TCP三次握手的安全漏洞对网络进行破坏的一种典型的拒绝服务攻击技术,它通过控制僵尸机向目标机器发送大量伪造IP的SYN报文,致使三次握手的过程无法完成,目标机器的TCP半连接队列被迅速占满而不能及时释放,正常的TCP连接无法建立,最终导致目标机器崩溃[2]。目前SYN Flood防御研究虽已取得很多成果,然而没有很好的办法彻底解决SYN Flood攻击问题。本文在分析了现有SYN Flood防御模型的基础上,基于Linux平台,研究了新的防御模型,该模型能在高强度的SYN Flood攻击状态下创建正常的TCP连接。

1 现有防御模型分析

目前现有的SYN Flood防御模型主要有SYN Cookie、SYN Gateway和SYN Proxy。

(1)SYN Cookie模型的思想是根据SYN报文中字段的信息和特殊算法生成序列号来唯一标识与客户端的第一次握手,在第三次握手时检验该序列号[3]。该模型解决了TCP半连接队列容量有限的问题,但在SYN Flood攻击强度很大时,计算所有SYN包的标识序列号会消耗大量的CPU资源,使计算机运行缓慢,甚至导致死机。

(2)SYN Gateway模型的思想是根据服务器承受的TCP全连接数目要远远高于TCP半连接数目,在关键的第三次握手时防火墙代替客户端生成ACK报文发给服务器,提前创建TCP空连接,从而减少服务器中TCP半连接的数目[4]。SYN Gateway从某种程度上可以有效减轻SYN Flood对服务器的攻击,但当SYN Flood攻击强度加大时,服务器会产生大量的空连接,若没有及时释放,最终服务器TCP连接数目会达到上限,从而造成拒绝服务。

(3)SYN Proxy模型的思想是防火墙首先作为"服务器"通过三次握手与客户端建立TCP连接,再作为"客户端"通过三次握手与服务器创建TCP连接,然后作为中转站转发客户端与服务器之间的数据包[5],模型如图1所示。在该模型中防火墙完全代替服务器来抵御SYN Flood攻击,运行负载重,处理TCP半连接的数量仍然有限,很难防御高强度的SYN Flood攻击。

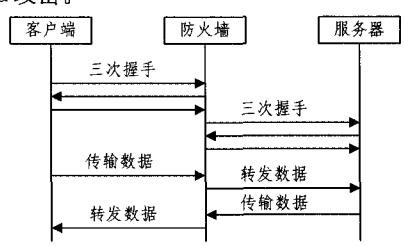

图1 SYN Proxy模型

本文受贵阳学院联合基金项目(黔科合J字LKG[2013]51号)资助。

刘 云(1981—),女,硕士,讲师,主要研究方向为计算机应用技术。

2 新的 SYN Flood 防御模型

深入分析 SYN Proxy 模型，客户端与服务器之间从创建 TCP 连接到传输数据，都有防火墙的参与，这带来了如下问题：防火墙经过了两次“三次握手”的过程，分别生成了与客户端和服务器对应的序列号与确认号，传输数据时，防火墙充当中转角色，需要不断匹配和修改每个数据包的序列号与确认号，此过程会消耗大量资源。同时，防火墙需要记录客户端与服务器的各种状态值，增加了负载。另外，第二个“三次握手”过程对客户端会引起 TCP 连接的延迟。

针对 SYN Proxy 模型的不足，研究并提出了新的防御机制，即增强的 SYN Proxy 模型，如图 2 所示。TCP 连接单元与客户端经过三次握手后创建连接，并将连接信息移交给 TCP 服务单元；TCP 服务单元充当服务器的角色，根据此连接直接为客户端提供服务，无需 TCP 连接单元的参与，TCP 连接单元只对创建连接的数据报文进行过滤。对比图 1 与图 2，增强的 SYN Proxy 模型消除了创建 TCP 连接的延迟和中转数据的资源负载。

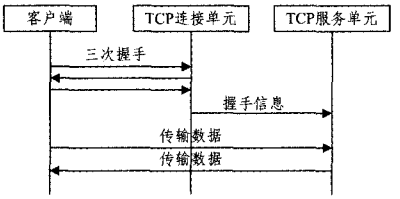

图 2 增强的 SYN Proxy 模型

3 模型实现和相关算法

增强的 SYN Proxy 模型实现包括两部分：防御 SYN Flood 攻击和 TCP 连接的移交。TCP 连接单元是防御 SYN Flood 攻击的关键部分，它检测攻击的强度，采用改进的一种综合算法进行防御，创建正常的 TCP 连接；基于 Linux 平台，修改 TCP 内核模块，使 TCP 服务单元在没有经过“三次握手”的过程下，能通过移交的连接信息直接创建 TCP 连接。

3.1 SYN Flood 攻击检测

SYN Flood 攻击的主要特征：一是 SYN 包数量巨大，二是每一个 SYN 包中的 IP 是伪造的。因此很难在第一次握手时从 SYN 报文的静态特征中分析出该数据包是攻击包还是正常包，可以通过 SYN 包的流量、第二次握手反应及 TCP 半连接的数量等动态数据来检测 SYN Flood 攻击的强度。

设 TCP 半连接队列容量为 M，TCP 半连接数目为 N，单位时间(可实时调整)内新产生的 TCP 半连接数为 X，单位时间内因未从 TCP 半连接转换成 TCP 连接而出队的数目为 F，判断 SYN Flood 攻击强度如图 3 所示(代码实现略)。TCP 半连接队列容量 M 由 sys_ctl_max_syn_backlog 和 listen()传入的长度决定。

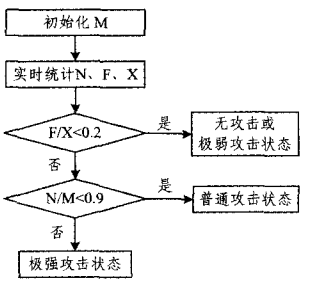

图 3 判断 SYN Flood 攻击强度

3.2 SYN Flood 攻击防御

文献[3]发现，在一次 SYN Flood 攻击中，约有 0.6%～14%的 IP 重复出现，在一次正常大流量中，约有 82.9%的 IP 曾发送过请求。基于对历史 IP 的行为参考，设计 IPstate 结构，用于存储历史 IP 状态(正常或异常)。IPstate 每个元素占用一位的宽度，元素存储地址与 IP 形成映射关系，因此 IPstate 具有较高的查询和更新效率。在不同的 SYN Flood 攻击强度下采用不同的防御策略(代码实现略)，如下：

(1)无攻击或极弱攻击状态

防御策略如图 4 所示，为准确识别正常 IP，向客户端回复两次 SYN ACK 包，且在 IPstate 结构中只记录成功创建连接的正常 IP，忽略未成功创建连接的 IP。时间 T 通过策略统计引入对客户端回复 ACK 包时间学习机制来同步更新，避免设置静态值引起对正常 IP 的漏报。在 Linux 的 TCP 内核模块中，TCP 半连接保存于 syn_table 队列结构中，根据 TCP 半连接的标示将元素从 syn_table 结构中移除即可完成丢弃 TCP 半连接的操作。

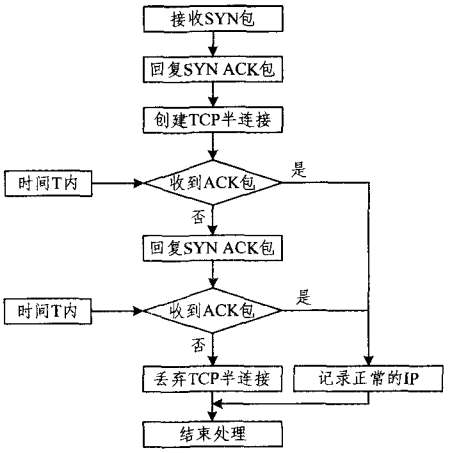

图 4 SYN Flood 极弱攻击状态下防御策略

(2)普通攻击状态

防御策略如图 5 所示。在 SYN Flood 普通攻击状态下，参考历史 IP 的行为，若是异常 IP，则直接丢弃 SYN 报文；syn_table 队列中的元素数量逐渐增大，对客户端的连接请求只回复一次 SYN ACK 包；在 IPstate 结构中只记录未知 IP 的异常或正常状态，对于正常状态的 IP，若未成功创建连接则忽略。

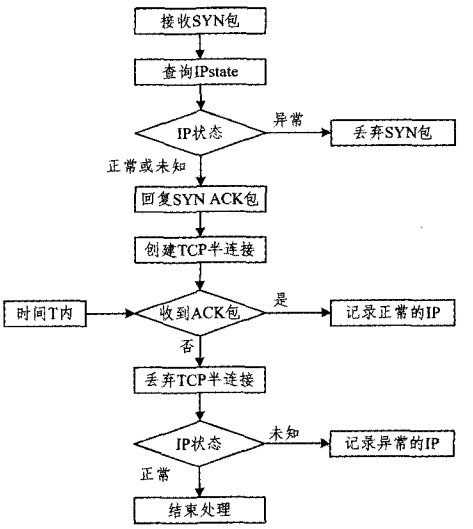

图 5 SYN Flood 普通攻击状态下防御策略

(3)极强攻击状态

防御策略如图 6 所示。在 SYN Flood 极强攻击状态下，通过 SYN Cookie 机制处理 SYN 报文，为避免计算 Cookie 造成的 CPU 资源负载，直接丢弃未知 IP 发送的 SYN 包。正常发送的 SYN 报文若没有得到响应，会再次发送[6]，而异常 SYN 报文则不会，因此针对性的丢包算法不会使正常 TCP 连接建立受到影响，而且大大降低了计算异常 SYN 报文 Cookie 的几率。设计类似 IPstate 的 IPthrow 结构，存储丢弃的 SYN 报文中的 IP。

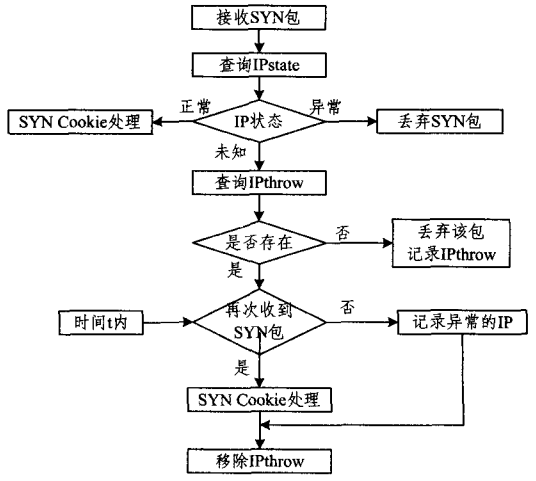

图 6　SYN Flood 极强攻击状态下防御策略

3.3　移交 TCP 连接

Linux 是一个自由开源的操作系统，网络协议栈的实现嵌在系统内核中，通过对 TCP 内核定制开发可满足网络通信场景的需求[7]。通过研究和学习 Linux TCP 内核源代码，对其做如下修改。首先，定义记录握手信息的结构体：

```
struct handinfo
{
  __u32 cip;/*客户端 ip*/
  __u16 cport;/*客户端端口号*/
  __u16 sport;/*服务器端口号*/
  __u32 seq;/*ACK 包的序列号*/
  __u32 ack;/*ACK 包的确认号*/
  __u32 size;/*ACK 包的大小*/
  __u16 window;/*窗口大小*/
  __u16 backup;/*保留*/
};
```

并定义存放握手信息的队列：

```
static struct handinfo
*handinfo_table[HANDINFO_QUEUE_LENGTH];
```

(1)tcp_v4_do_rcv()是 TCP 模块接收数据的入口函数，通过调用 tcp_v4_hnd_req()检查三次握手是否完成，若完成，则调用 tcp_check_req()创建 socket，存入 icsk_accept_queue 队列结构。修改 tcp_v4_hnd_req()和 tcp_check_req()，在三次握手完成时从 ACK 报文中分析出握手信息，存于 handinfo_table 队列结构。

(2)向 Socket 层提供 acceptHandinfo()接口，并通过添加的 handinfo_read()函数监视 handinfo_table 队列，在队列中有元素产生时便唤醒阻塞的 acceptHandinfo()。

(3)应用层程序调用 acceptHandinfo()，获取握手信息，基于 TCP 协议发送到 TCP 服务单元。

(4)TCP 服务单元修改 tcp_v4_do_rcv()函数，增加对接收报文的判断，若是 SYN/ACK 报文，仍通过 tcp_v4_hnd_req()函数经过三次握手创建连接；若是握手信息报文，则调用添加的 tcp_conn_hand()函数，该函数根据握手信息生成 socket，将其状态置为 TCP_ESTABLISHED 后存入 icsk_accept_queue 队列结构。应用层程序仍通过 accept()获取建立连接的 socket[8]。

4　防御测试

搭建增强的 SYN Flood 模型防御测试环境，测试模型如图 7 所示。TCP 连接单元与 TCP 服务单元使用 PC 机(1 个四核 3.30G CPU，4G DDR2 内存，160G SATA 硬盘，Linux 2.6系统)；SYN Flood 攻击客户端使用 30 台计算机，IP 从 192.168.10.21 到 192.168.10.50，由攻击控制端发送指令开始模拟攻击，攻击工具使用 Stacheldraht；正常客户端使用 5 台计算机，IP 从 192.168.10.60 到 192.168.10.65。

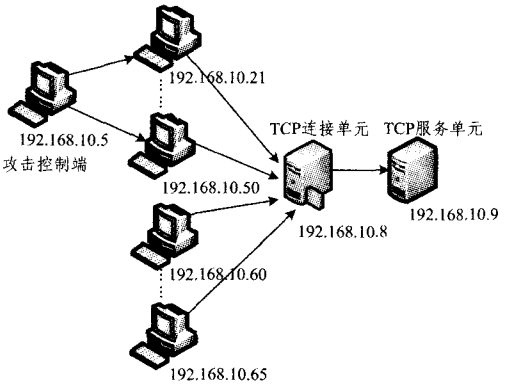

图 7　测试模型

测试分两步进行：(1)开启 TCP 连接单元的防火墙功能，通过 SYN Proxy 机制防御 SYN Flood 攻击，统计成功创建的 TCP 连接数目；(2)关闭 TCP 连接单元的防火墙功能，通过增强的 SYN Proxy 机制防御 SYN Flood 攻击，统计成功创建的 TCP 连接数目。测试结果如表 1 所列。

表 1　建立 TCP 连接数目统计表

SYN Flood 攻击强度(秒)	SYN Proxy 模型	增强的 SYN Proxy 模型
10000	98.2%	98.3%
15000	86.8%	87.1%
20000	42.1%	83.5%
25000	3.2%	80.3%
30000	0.6%	40.2%
40000	0%	25.6%

分析表 1 数据，通过 SYN Proxy 机制防御攻击，在攻击强度超过 15000 时，成功创建的 TCP 连接数目急剧下降，当强度超过 25000 时，几乎不能建立正常的 TCP 连接；通过增强的 SYN Proxy 机制防御攻击，在攻击强度超过 25000 时，成功创建的 TCP 连接数目开始下降，当强度到 40000 时，仍有 1/4 的 TCP 连接成功创建。由此可见，增强的 SYN Proxy 模型较之 SYN Proxy 模型能有效防御高强度的 SYN Flood 攻击，并提供正常的 TCP 服务。

结束语　在本文提出增强的 SYN Proxy 模型中，TCP 连接单元根据检测的 SYN Flood 攻击强度综合运用历史 IP 行为参考、SYN Cookie、针对性丢包算法等方法采用相应防御策略，将建立的 TCP 连接移交给 TCP 服务单元，TCP 服务单

元修改 Linux TCP 内核模块，使其在没有经过三次握手的过程下可以直接创建 TCP 连接。经仿真测试表明，增强的 SYN Flood 模型能更有效地防御高强度的 SYN Flood 攻击，并提供正常的 TCP 服务。基于 Linux 平台，增强的 SYN Proxy 模型实现简单，部署方便，较之目前现有的防御模型有更好的优越性。

参 考 文 献

[1] 一江水. TCP 协议三次握手过程分析[EB/OL]. http://www.cnblogs.com/rootq/articles/1377355.htm,2013-01-05

[2] 李蓬. DDoS 攻击原理及其防御机制的研究[J]. 通信技术，2010,43(4):96-98

[3] 胡鸿，袁津生，郭敏哲. 基于 TCP 缓存的 DDoS 攻击检测算法[J]. 计算机工程，2009,35(16):112-114

[4] 曾小荟，冷明，刘冬生，等. 一个新的 SYN Flood 攻击防御模型的研究[J]. 计算机工程与科学，2011,33(4):35-39

[5] 赵广利，江杨. Linux 平台下防御 SYN Flood 攻击策略的研究[J]. 计算机工程与设计，2009,30(10):2394-2397

[6] 徐图，何大可，邓子健. 分布式拒绝服务攻击特征分析与检测[J]. 计算机工程与应用，2007,43(29):146-149

[7] 王海花，杨斌. Linux TCP/IP 协议栈的设计及实现特点[J]. 云南民族大学学报：自然科学版，2007,16(1):73-76

[8] 赵国锋，邱作雨，张毅. 基于单片机的嵌入式 TCP/IP 协议栈的设计与实现[J]. 计算机技术与发展，2010,19(3):137-140

(上接第 183 页)

高，通信开销较小，能较好地完全量子组密钥的服务，对其大规模服务用户有一定的理论指导意义。后续研究中，将完善组播成员动态变化时密钥的更新环节，并且逐步将研究重点投入到存在中继节点的 QKD 组网环境中，从而提出一套功能更为完善、应用更为广泛的量子组密钥服务方案。

参 考 文 献

[1] Patrick P, John C, David K. Distributed Collaborative Key Agreement Protocols for Dynamic Peer Grou-ps[C]// Computer Science Technical Reports. 2002:02-015

[2] Yongdae K, Adrian P, Gene T. Group Key Agreement Efficient in Communication[C]// IEEE Transactions on Computers. 2003:19-57

[3] 张江，张萌，陈春晓，等. 高效的分布式组密钥协商机制[J]. 清华大学学报，2008,48(1):101-105

[4] 张玉臣，王亚弟，韩继红，等. 自组网环境下基于组合公钥的分布式密钥管理[J]. 计算机科学，2011,38(10):75-77

[5] 赵秀凤，徐秋亮，韦大伟. 群组密钥协商协议的安全性分析方法研究[J]. 计算机科学，2011,38(6):145-148

[6] 陈卫东，刘广伟，刘泽超，等. 分布式组播密钥管理协议中的组密钥生成算法研究[J]. 小型微型计算机系统，2010,31(7):1307-1310

[7] 刘成林，徐秋亮. 基于身份的多安全群组密钥协商协议[C]// 济南：第九届中国密码学学术会议论文集. 2006

[8] 赵龙泉. 基于密钥树的组密钥更新技术研究[D]. 郑州：解放军信息工程大学，2010

[9] 刘广伟. 安全组播中的组密钥管理协议研究[D]. 沈阳：东北大学，2009

(上接第 195 页)

表 4 DPS 的可信性评估结果

	需求分析	软件设计	编码实现	软件测试	多阶段融合后
可用性	0.909	0.933	0.901	0.839	**0.905**
实时性	0.885	0.876	0.888	0.839	**0.839**
可靠性	0.787	0.797	0.731	0.611	**0.792**
安全性	0.768	0.766	0.784	0.786	**0.798**
可生存性	0.679	0.732	0.760	0.644	**0.749**
效能性	0.758	0.744	0.766	0.795	**0.795**
可维护性	0.716	0.625	0.752	0.853	**0.714**
		可信性			0.799

作为关键软件，DPS 经过多次工程任务的考验并得到了用户好评，表 4 所列的评估结果较为符合该软件的实际情况，证明了所提全生命周期软件可信性评估方法的有效性。

结束语 软件质量度量与评估是一个重要而又困难的研究课题，是软件工程中迫切需要解决的一个难题。本文提出的全生命周期软件可信性评估模型综合采集生命周期各阶段的可信度量数据，设计的基于数据融合理论的定量评估算法能有效处理多阶段多类型多量纲的数据并进行合理推理，使用的基于知识发现的权值获取方法可以有效降低评估过程中的主观性。最后，工程实践证明该方法能够给出较为准确的定量评估结果。

下一步将就如何改进可信度量指标的设计，以更全面有效地采集软件全生命周期各阶段的度量数据进行更加深入的研究。

参 考 文 献

[1] 刘克，单志广，王戟，等. 可信软件基础研究重大研究计划综述[J]. 中国科学基金，2008,22(3):145-151

[2] McCall J. The Automated Meaz of Software Quality[C]// 5th COMMPSAC. 1981

[3] ISO/IEC 9126 Information Technology—Software Product Evaluation—Quality Characteristics and Guidelines for Their Use. First Ed, Dec. 1991

[4] Zhang Wei-xiang, Liu Wen-hong, Du Hui-sen. A Software Quantitative Assessment Method Based on Software Testing[C]// Lecture Notes in Artificial Intelligence (LNAI) 7390. Springer, 2012:300-307

[5] 王胜芝，鲜明，王雪松，等. 软件质量综合评价方法研究[J]. 计算机工程与设计，2002,23(4):16-18

[6] 董剑利，时宁国. 基于软件质量评估的模糊综合评判算法研究与改进[J]. 计算机工程与科学，2007,29(1):66-68

[7] 杨善林，丁帅，褚伟. 一种基于效用和证据理论的可信软件评估方法[J]. 计算机研究与进展，2009,46(7):1152-1159

[8] 康耀红. 数据融合理论与应用[M]. 西安：西安电子科技大学出版社，1997

[9] 李烨，蔡云泽，尹汝泼，等. 基于证据理论的多类分类支持向量机集成[J]. 计算机研究与发展，2008,45(4):571-578

[10] Shafer G. A Mathematical Theory of Evidence[M]. Princeton U P, Princetion, 1976

[11] Yager R R. On the D-S framework and new combination rules [J]. Information Sciences, 1987,41(2):93-138

[12] 孙全，叶秀清，顾伟康. 一种新的基于证据理论的合成公式[J]. 电子学报，2000,28(8):117-119