文章编号：1007-5321(2017)增-0077-04 DOI:10.13190/j.jbupt.2017.s.017

Web 服务器应用层慢速 DDoS 攻击防御研究

周椿入[1]， 刘晓明[2]， 雷 敏[1]， 武旭东[3]， 邓诗琪[1]
(1. 北京邮电大学 信息安全中心，北京 100876；
2. 国家计算机网络应急技术处理协调中心，北京 100029；
3. 四川科瑞软件有限责任公司，四川 绵阳 621000)

摘要：DDoS 攻击会导致 Web 服务器无法向用户提供正常的服务. 应用层 DDoS 攻击不同于网络层 DDoS 攻击，所有应用层 DDoS 请求都是合法的. 慢速 DDoS 攻击主要利用的是 thread-based 架构的服务器的特性，这种服务器会为每个新连接打开一个线程. 攻击者和 Web 服务器建立正常的 HTTP 连接以后，通过各种方法保持这个连接，从而占用服务器大量的资源. 对应用层慢速 DDoS 的原理进行分析，并提出了相应的防御方法，能提高服务器抗 DDoS 攻击的能力，从而有效地提升服务器的安全性能.

关 键 词：Web 服务器；慢速 DDoS 攻击；安全防御
中图分类号：TN911.22 **文献标志码**：A

Research on Defense of Slow DDoS Attack on Web Server Application Layer

ZHOU Chun-ru[1], LIU Xiao-ming[2], LEI Min[1], WU Xu-dong[3], DENG Shi-qi[1]
(1. Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China;
2. National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China;
3. Sichuan Kerui Software Co. Ltd, Sichuan Mianyang 621000, China)

Abstract: Web server can't provide normal service for users under DDoS attack. DDoS attack on application layer is different from DDoS attack on network layer, and each request sent by DDoS attack on application layer is legal. Slow DDoS attack mainly uses thread-based architecture characteristics of Web server. Web server of this type will open a new thread for every new connection. After the attacker has built a normal HTTP connection with Web server successfully, he will hold this connection through all kinds of methods to take up a lot of resources of Web server. The principle of slow DDoS on application layer is analyzed and related defense methods are proposed, which can improve ability to resist DDoS attack of Web server and safety performance.

Key words: Web server; slow DDoS attack; security defense

分布式拒绝服务(DDoS,distribute deny of service)攻击会导致 Web 服务器无法向用户提供正常的服务. 应用层的 DDoS 攻击和网络层的 DDoS 攻击不同，应用层的 DDoS 攻击的请求全部是合法业务请求. Web 服务器应用层的 DDoS 攻击方法是通过大量消耗 Web 服务器的正常资源，从而导致 Web 服务器无法向用户提供正常的服务. 为了保护 Web 信息系统的安全，需要对 Web 信息系统进行安全分析与评估. 慢速 DDoS 攻击主要利用的是 thread-based 架构的服务器的特性，这种服务器会为每个新

收稿日期：2016-05-26
基金项目：国家科技支撑计划课题(2015BAH08F02)
作者简介：周椿入(1993—)，男，硕士生，E-mail：zcr214@butp.edu.cn；刘晓明(1979—)，男，高级工程师.

连接打开一个线程. 攻击者和 Web 服务器建立正常的 HTTP 连接以后,通过各种方法保持这个连接,从而大量占用服务器的资源.

通过搭建 Web 服务器模拟系统对常见慢速 DDoS 攻击原理进行分析,从应用层 DDoS 攻击的特点出发,研究并提出应用层慢速 DDoS 攻击的防御方法,提高服务器抗 DDoS 攻击的能力,从而提高服务器的安全性能.

介绍了慢速 DDoS 的攻击原理,对慢速 DDoS 攻击原理实验进行了描述并提出应对慢速 DDoS 攻击的防御方法.

1 慢速 DDoS 原理

应用层 DDoS 攻击大量消耗 Web 服务器的正常资源,从而导致 Web 服务器无法向用户提供正常服务而且应用层 DDoS 攻击不同于网络层 DDoS 攻击,所有应用层 DDoS 攻击的请求都是合法的. 其中最为典型的攻击为 CC 攻击. CC 攻击的本名为 HTTP-FLOOD,是一种专门针对 Web 服务器的应用层 FLOOD 攻击,攻击者操纵网络上的被控肉鸡,对攻击目标 Web 服务器进行海量 http request 攻击,直到服务器带宽被占满,无法向其他用户提供正常的访问,造成了拒绝服务[1]. 由于伪造的 HTTP 请求和客户正常请求没有区别,加大了防御的难度.

CC 攻击需要攻击者控制网络上大量的肉鸡计算机,利用这些被控肉鸡计算机在短时间内向被攻击 Web 服务器发送海量攻击请求. 慢速 DDoS 攻击无须控制网络上其他肉鸡计算机,只需要使用攻击者一台个人计算机就可以让一个中小型 Web 服务器在 1 min 之内拒绝其他用户的服务,从而无法提供 Web 服务.

慢速 DDoS 攻击的原理如下:对任何一个开放了 HTTP 访问的 Web 服务器,攻击者先建立一个连接,指定一个比较大的 Content-Length,然后以非常低的速度发包,比如 1 ~ 10 s 发一个字节,然后一直维持该连接不断开. 如果客户端持续建立多个此种类型的连接,Web 服务器可用的连接将逐步被攻击者占满,从而 Web 服务器无法向正常用户提供服务,导致拒绝服务[2].

慢速 DDoS 攻击分为 3 种类型.

1) Slow headers. 因为 HTTP 头部包含了一些 Web 应用可能用到的重要信息,Web 应用服务器必须在接收完所有的 HTTP 头部以后才能处理 HTTP 请求中的数据. Web 服务器必须收到 2 个连续的 \r\n时才会认为 HTTP 头部发送完毕. 攻击者利用这个特点,向 Web 服务器发起一个 HTTP 请求,一直不停地发送 HTTP 头部,消耗服务器的连接和内存资源. 攻击方法是:攻击者客户端与服务器建立 TCP 连接后,每 30 s 才向服务器发送一个 HTTP 头部,因为 Web 服务器一直没有收到 2 个连续的 \r\n,Web 服务器认为客户端没有发送完头部,从而持续等待客户端发送数据.

2) Slow body. 攻击者向 Web 服务器发送一个 HTTP POST 请求,该请求的 Content-Length 头部值很大,Web 服务器以为客户端要发送很大的数据. 服务器会一直保持连接准备接收数据,但攻击客户端每次只发送很少量的数据,使该连接一直保持存活,消耗服务器大量资源. 攻击方法是:攻击客户端与服务器建立 TCP 连接后,发送完整的 HTTP 头部,但是 HTTP POST 方法带有较大的 Content-Length,然后每 10 s 发送一次随机的参数. Web 服务器因为没有接收到相应 Content-Length 的 HTTP 报的内容,而持续地等待客户端发送数据,造成 Web 服务器资源被无效连接所占用,无法向客户提供正常服务[3-4].

3) Slow read. 攻击者客户端与 Web 服务器建立连接并发送了一个 HTTP 请求,客户端发送完整的请求给服务器端,然后一直保持该连接,以很低的速度读取 Web 服务器的 Response,比如很长一段时间攻击者客户端不读取任何数据,通过发送 Zero Window 到服务器,让服务器误以为客户端很忙,直到连接快超时之前才读取一个字节,以消耗服务器的连接和内存资源. 攻击方法是:客户端把数据发给服务器后,服务器发送响应时,收到了客户端的 Zero Window 提示(表示自己没有缓冲区用于接收数据),服务器不得不持续地向客户端发出 Zero Window Probe 包,询问客户端是否可以接收数据[5-6].

以 3 种类型中的 Slow body 慢速 DDoS 攻击为主要研究对象.

2 仿真实验

2.1 环境搭建

本系统所使用的实验环境分为两部分,其中一部分为搭建了 Web 服务器模拟系统的靶机,Web 服务器为使用了 thread-based 架构的 Apache;另外一部分是攻击机,其中攻击机的系统环境为已安装好攻击工具 Slow Http Test 的 Kali.

通过使用 Slow Http Test 工具模拟 HTTP 攻击并向系统发起攻击使 Web 服务器不断保持连接等待状态并且消耗资源，使用服务器压力测试工具 httperf 和 autobench 测试出在服务器达到处理请求的门限值时，相应的连接速率、请求速率等不再变化.

2.2　Slow Http Test

Slow Http Test 是一款对 Web 服务器进行慢攻击的测试软件，慢攻击是相对于 CC 或者 DDoS 的快而言的，并不是只有量大速度快才能使 Web 服务器崩溃，使用慢攻击有时候也能达到同一效果. Slow Http Test 测试软件包括 Slowloris，Slow HTTP POST，Slow Read Attack 等攻击方式. 这些慢攻击工具的原理就是让 Web 服务器等待，让服务器在保持连接等待时，不断消耗资源.

本实验中使用的慢攻击方式为 Slow HTTP POST，在 POST 的提交方式中，允许在 HTTP 的头中声明 content-length，也就是 POST 内容的长度. 在提交了头以后，将后面的 body 部分卡住不发送，这时服务器在接受了 POST 长度以后，就会等待客户端发送 POST 的内容，攻击者保持连接并且以 10 ~ 100 s一个字节的速度去发送，就达到了消耗资源的效果，因此不断地增加这样的连接，就会使得服务器的资源被消耗，最后可能死机，但是可能对很多 Web 服务器程序已经失效了[7-8].

2.3　仿真结果

通过使用 Slow Http Test 工具对靶机发起慢速 DDoS 攻击，并用 httperf 和 autobench 压力测试工具对服务器进行监测，得到仿真数据如表 1 所示. 其中，1）R_r 为每秒向 Web 服务器发送的请求数（并发数）；2）R_c 为每秒建立连接的请求数；3）R_{min} 为每秒响应的最小请求数；4）R_a 为每秒响应的平均请求数；5）R_{max} 为每秒响应的最大请求数；6）T 为服务器响应时间. 将表 1 数据绘成相应图形便于直观分析，得到结果如图 1 所示.

表 1　步长为 10 时的压力测试仿真数据

并发数	R_c	R_{min}	R_a	R_{max}	T
10	9.8	9.0	9.8	10.4	571.5
20	18.8	18.6	19.1	19.6	424.9
30	27.2	27.0	27.0	27.0	457.5
40	31.6	35.4	35.4	35.4	529.2
50	38.0	37.8	37.8	37.8	810.8
60	38.3	38.0	38.0	38.0	1 115.3
70	38.2	38.0	38.0	38.0	1 330.3
80	35.5	38.4	38.4	38.4	1 478.8
90	37.2	37.6	37.6	37.6	1 721.4
100	35.4	38.2	38.2	38.2	1 701.5
110	30.6	39.0	39.0	39.0	1 802.7
120	34.5	37.6	37.6	37.6	1 962.8
130	39.1	38.8	38.8	38.8	1 947.3
140	38.9	39.0	39.0	39.0	2 005.7
150	38.9	38.8	38.8	38.8	2 083.5

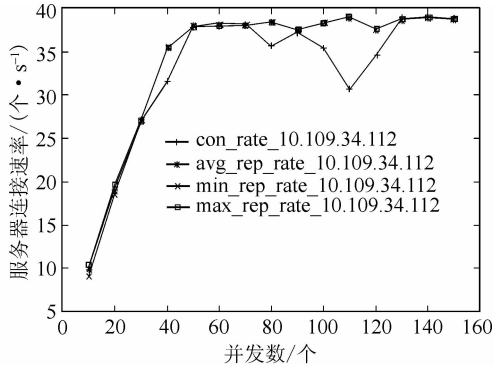

图 1　并发数 10 ~ 150（步长为 10）

结合表 1 和图 1 可以看出，在并发数小于等于 50 前，随着并发数不断增加，服务器的连接速率几乎相应地增长，每秒响应请求数的各项指标 R_{min}、R_a、R_{max} 也不断增加，服务器的响应时间值变化不大，比较稳定，但在并发数达到 50 之后，服务器的连接速率近似不变（因网络不稳定存在小波动），每秒响应请求数的各项指标 R_{min}、R_a、R_{max} 也逐渐趋于稳定，服务器的响应时间值突然大幅度增加，并一直趋于增长趋势. 实验结果表明，服务器处理响应请求的门限值近似为 50，在慢速 DDoS 攻击的并发数达到 50 之后，服务器已达到请求最大承受能力.

3　慢速 DDoS 攻击防御方法

3.1　使用 Anti-DDoS 设备

针对 HTTP 慢速 DDoS 攻击的特点，Anti-DDoS 设备对每秒钟 HTTP 并发连接数进行检查，当每秒钟 HTTP 并发连接数超过设定值时，会触发 HTTP 报文检查，检查出以下任意一种情况，都认定受到 HTTP 慢速连接攻击，则将该源 IP 地址判定为攻击源，加入动态黑名单，同时断开此 IP 地址与 HTTP 服务器的连接.

1）连续多个 HTTP POST 报文的总长度都很大，但是其 HTTP 载荷长度都很小.

2）连续多个 HTTP GET/POST 报文的报文头都没有结束标识.

3.2　Nginx

Nginx（engine x）是一个高性能的 HTTP 和反向

代理服务器,也是一个 IMAP/POP3/SMTP 服务器. 其特点是占有内存少,并发能力强,事实上 Nginx 的并发能力确实在同类型的网页服务器中表现较好[9]. 利用搭建好的 Nginx 服务器,主要有两种防御方法可以有效抵抗慢速 DDoS 攻击:主动防御法和被动防御法.

1) 主动防御法

为了让 Nginx 支持更多的并发连接数,根据实际情况对工作线程数和每个工作线程支持的最大连接数进行调整. 例如设置"worker_processes 10"和"worker_connections 1024",那这台服务器支持的最大连接数就是 10 × 1 024 = 10 240. Nginx 0.7 开始提供了 2 个限制用户连接的模块:NginxHttpLimitZoneModule 和 NginxHttpLimitReqModule. NginxHttpLimitZoneModule 可以根据条件进行并发连接数控制. Nginx Http Limit Req Module 可以根据条件进行请求频率的控制[10].

2) 被动防御法

① 禁止 IP 地址

访问者通过浏览器正常访问网站,与服务器建立的连接一般不会超过 20 个,可以通过脚本禁止连接数过大的 IP 访问.

② 根据特征码屏蔽请求

一般同一种慢速 DDoS 攻击工具发起的攻击请求包总是相同的,而且和正常请求有所差异. 当服务器遭遇慢速 DDoS 攻击时,可以快速查看日志,分析其请求的特征,比如 User-agent,将攻击请求包中的参数值作为特征进行过滤,将符合特征的请求全部拒绝访问[11].

4 结束语

慢速 DDoS 攻击主要利用的是 thread-based 架构的服务器的特性,这种服务器会为每个新连接打开一个线程. 攻击者和 Web 服务器建立正常的 HTTP 连接以后,通过各种方法保持这个连接,从而大量占用服务器的资源. 从应用层 DDoS 攻击的特点出发,通过搭建 Web 服务器模拟系统对常见慢速 DDoS 攻击原理进行分析,并提出应用层慢速 DDoS 攻击的防御方法,能提高服务器抗 DDoS 攻击的能力,从而提高服务器的安全性能. 在未来的研究工作中,将继续通过仿真实验对提到的慢速 DDoS 攻击的防御方法进行检测,并验证其抗慢速 DDoS 攻击的能力.

参考文献:

[1] 徐琳. 应用层 DDoS 攻击防御与检测方法[D]. 上海:上海交通大学, 2013.

[2] 田开琳. 低速 DDoS 攻击的异常检测[D]. 上海:华东师范大学, 2010.

[3] 冯鸿雁, 刘利锋. 一种防止网络攻击的方法及装置[M]. 2008.

[4] 王飞. 分布式拒绝服务攻击检测与响应技术研究[D]. 长沙:国防科学技术大学, 2013.

[5] 张永铮, 肖军, 云晓春, 等. DDoS 攻击检测和控制方法[J]. 软件学报, 2012, 23(8):2058-2072.
Zhang Yongzheng, Xiao Jun, Yun Xiaochun, et al. DDoS attacks detection and control mechanisms[J]. Journal of Software, 2012, 23(8):2058-2072.

[6] 王浩. 针对 TCP 的低速 DDoS 解析及防御策略[J]. 计算机工程, 2009, 35(13):122-124.
Wang Hao. Low-rate TCP-targeted DDos analysis and defense Policy[J]. Computer Engineering, 2009, 35(13):122-124.

[7] 余双成. DDoS 攻击检测技术研究[D]. 北京:北京邮电大学, 2013.

[8] 张烜. 基于应用层的 DDoS 攻击检测防御技术研究[D]. 北京:北京邮电大学, 2009.

[9] 田纯青. 利用 Nginx 实现基于 URI 的 Web 负载分配[J]. 现代计算机(专业版), 2009(7):187-191.
Tian Chunqing. Using Nging to implement web load distribution based on URI[J]. Modern Computer, 2009, (07):187-191.

[10] 吴迪, 徐国胜. 一种基于 Nginx 的安全设备代理方案[C]// 中国通信学会学术年会. 2012.

[11] 吴迪. 基于 Nginx 的安全管理系统的设计与实现[D]. 北京:北京邮电大学, 2013.