文章编号:1001 - 9081(2009)03 - 0826 - 04

代理服务器特性探测技术研究

杨 杰,舒 辉
(信息工程大学 信息工程学院,郑州 450002)
(baggiojj@ hotmail. com)

摘 要:从用户角度远程探测代理服务器的通信特性。对代理的功能及分类进行总结,提出了代理服务器工作模型,对其逻辑模块和影响通信的因素进行分析。在此基础上分析代理特性探测原理,以 HTTP 代理为例设计了探测算法并实现了探测系统原型。该系统实现了 Windows 域环境下特定认证机制的穿透;经过测试并分析其性能,具有一定的适用性。

关键词:代理服务器;探测;HTTP 代理;认证
中图分类号: TP393.09 **文献标志码**:A

Research on probing characteristics of proxy server

YANG Jie, SHU Hui
(*College of Information Engineering, Information Engineering University, Zhengzhou Henan* 450002, *China*)

Abstract: The communication characteristics of proxy server were investigated from the angle of remote host computer. The functions and classifications of proxy server were summarized, and its working model was proposed, together with analysis of its logic modules and factors that influenced communication. After the principle of probing characteristics of proxy server was analyzed, a probing algorithm for HTTP proxy was designed, and a probing system prototype was implemented. The system accomplished penetrating through specific authentication mechanism in Windows domain environment. At last, the system was tested and its performance was analyzed. It is concluded that it is applicable to a certain extent.

Key words: proxy server; probe; HTTP proxy; authentication

0 引言

代理服务器是用户与 Internet 服务提供者之间的一种中间代理机构,通常由软件实现,用于转发网络通信数据,并对转发过程进行控制和记录[1]。随着网络的迅速发展,代理服务器被广泛使用。常见的代理软件就有十多种,可对其作不同的配置以实现不同的功能。

代理服务器通常会对用户的网络通信进行访问控制,而用户主机上的应用程序对代理服务器的配置状况往往并不了解,若这些程序的网络通信不能适应当前的网络环境,会造成通信效率低下甚至阻碍正常的数据交互。特别是一些新兴的网络服务(例如 P2P 文件共享、即时通信等)需要在复杂网络环境下进行数据交互,代理服务器无疑是影响通信的重要方面。如果能够远程探测代理服务器的配置情况,就能有针对性地进行网络通信,提高通信效率;在非授权条件下,获取代理服务器特性将直接关系到能否正常通信。

因此,对代理服务器的特性进行探测很有必要,而这需要对代理服务器影响网络通信的诸多特性进行分析总结,并研究探测这些特性的技术。

1 相关研究

早期对代理服务器的探测多是通过扫描来探测代理是否存在及其开放的端口号,也出现了实现相应功能的代理扫描软件。这些对代理特性的探测比较粗略,其目的只是寻找可用的代理服务器。本文所要讨论的代理服务器特性探测是指获取代理服务器的属性信息,从而保证通信可行性并提高通信效率,是一种更为精确的探测。

当前关于代理服务器的理论研究主要集中在从服务器的角度优化代理性能[2],例如缓存策略、代理的网络拓扑位置、多代理协同等;而从用户的角度研究代理服务器通信特性的情况不多,仅有一些针对某种具体代理软件(例如 ISA、Squid)的特性研究或配置方法介绍[3-4],以及几种代理软件的性能比较。

2 代理服务器特性分析

2.1 代理服务器功能与分类

代理服务器是建立在 TCP/IP 协议应用层上的一种服务软件[1],主要代理网络用户去获取网络信息和资源。典型的代理服务器功能如表 1 所示。

代理服务器具有不同的工作模式,本文根据不同标准对其进行分类(如图 1 所示),以便理清代理特性,明确本文研究范围。

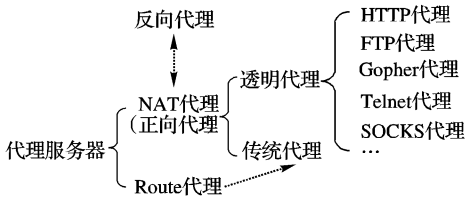

图 1 代理服务器分类

代理服务器转发数据有 2 种方式,一种是 NAT,用于内网

收稿日期:2008 - 09 - 16;**修回日期**:2008 - 11 - 01。

作者简介:杨杰(1981 -),男,湖北咸宁人,硕士研究生,主要研究方向:信息安全; 舒辉(1974 -),男,江苏盐城人,副教授,博士,主要研究方向:并行编译、信息安全。

主机访问 Internet,这是最普遍的情况;另一种是路由,在源和目标网络间传递数据,用于访问本机不能直接访问的网络资源,早期的代理探测即探测 Route 代理。另外,有些代理软件(如 CCProxy、Squid)可实现外网主机访问内网资源,即所谓的反向代理。

表 1　代理服务器功能表

功能	说明	作用
共享上网	让多台主机共享式地访问网络	降低上网成本,缓解现有 IPv4 合法 IP 地址不足的问题
缓存	缓存访问过的资源,当需访问相同资源且未过期时,直接从缓存中读取返回给提出请求的主机	提高访问网络的速度,减少 Internet 接入点的负载和流量,降低上网成本
防火墙	通常部署在内部局域网接入 Internet 的网关上,实现网络地址转换(NAT);通过访问控制规则过滤非法数据	使内外网不直接连接,对外屏蔽内网细节;为各种应用层服务提供安全保护
用户管理	管理通过其访问网络的用户,赋予不同用户相应的权限	提高安全性,兼顾整体处理能力和用户对性能的需求

根据是否需要用户设置软件的代理属性,NAT 代理可分为传统代理(Route 代理也属传统代理)和透明代理。传统代理需要在用户主机的 IE、Outlook Express 等软件设置代理服务器 IP 地址和端口等信息;而透明代理无需设置,用户感觉不到代理的存在,如 SyGate 和 Windows 自带的 Internet 连接共享均为典型的透明代理。透明代理通常不分析用户请求,访问控制和用户管理的功能很简单,一般也无缓存,响应速度快。

按照所代理的应用层协议,传统代理又可分为 HTTP 代理、FTP 代理、Gopher 代理、Telnet 代理、SOCKS 代理等。虽然协议不同,但实现原理相似,常见的代理软件大都支持对多种协议的代理。本文的研究对象即为传统代理(下文如不特别说明,均指传统代理)。

2.2　代理服务器工作模型

在研究常见代理软件特性的基础上,本文从用户的角度提出代理服务器工作模型(如图 2 所示)。该模型根据代理影响通信的层次,将其划分为通信数据转发、性能控制、访问控制和日志 4 个逻辑模块。

通信数据转发模块实现共享上网和部分的防火墙功能,它是代理服务器的通信模块。该模块决定了代理服务器转发数据的模式(NAT、Route 或反向代理)。

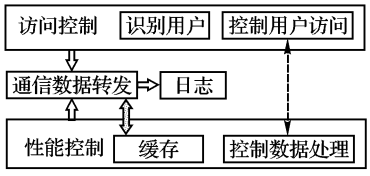

图 2　代理服务器工作模型

性能控制模块实现缓存和部分的用户管理功能,它从服务器性能的全局层面对通信数据进行控制。该模块可分为缓存和控制数据处理两部分。缓存部分与通信数据转发模块有数据交互,涉及到具体缓存策略的设定,旨在提高访问网络资源的效率,对合法的正常通信没有约束。控制数据处理部分主要包括是否支持持续连接、超时时间长度、最大并发连接数、最大数据包长度、数据类型等,这些因素直接关系到代理服务器的整体数据处理能力,作用于所有的合法用户,对其通信产生影响。

访问控制模块实现部分的防火墙和部分的用户管理功能,它从具体用户的局部层面对通信数据进行控制。该模块可分为识别用户和控制用户访问两部分,实质上分别对应认证和授权。识别用户部分通过 IP 地址、主机名、MAC 地址等属性,以及身份认证(包括自定义用户和基于 Windows 域的用户)来判定当前用户是否合法,对合法用户的正常通信没有约束。控制用户访问部分对合法用户进行访问控制,包括用户最大连接数、最大数据带宽、最大数据包长度、访问时间区间、协议(端口)类型、内容过滤(包括站点过滤、关键字过滤和数据类型过滤)等,不同用户被赋予不同的访问权限和通信能力。

日志模块对通信数据转发的处理过程进行记录,实际上体现的是性能控制与访问控制对通信过程的限制情况。该模块本身对通信没有直接影响,对用户也是透明的。

3　代理服务器特性探测

3.1　探测原理分析

代理服务器对用户通信的影响主要有两方面。一方面是判别用户身份,体现在访问控制模块的识别用户部分;另一方面是限制用户行为,又分为两个层次:首先是从代理服务器性能的全局角度对通信行为进行控制,体现在性能控制模块的控制数据处理部分;然后是从具体用户的局部角度对通信行为进行控制,体现在访问控制模块的控制用户访问部分。这两部分只是逻辑上针对不同对象,其实相互间有重叠和相似的地方,并没有严格地区分开来。

其中,识别用户部分直接关系到能否进行通信,必须首先进行。这部分相关属性较多,但通过直接尝试访问代理服务器,即可判定当前用户是否合法。如需身份认证,则应尽可能通过认证,否则后续探测无法进行。

控制数据处理部分是针对代理服务器整体性能的,如不加限制,可能会影响其正常工作。该部分涉及属性不多且相对固定,多数代理软件提供的相关设置项基本相似,其默认配置均对这些属性进行了限制。这些属性反映了代理服务器影响通信的基本特性,但由于探测程序不能独占代理服务器,对其中一些特性的探测是不可行的。因此,对能够探测的属性必须进行探测。

控制用户访问部分是在整体的性能控制基础上,对每个用户进行个性化的约束。该模块包含的属性更多更复杂,各种代理软件在实现上也存在差异;其默认配置通常不对这些属性作限制,需根据实际情况进行设置以提高安全性和效率。这些属性反映了代理服务器进一步影响通信的特性,是一种强化而非必须,对它们的探测是有可选的。

此外,鉴于探测的目的之一是提高通信效率,在获取代理特性的基础上,还应该探测应用程序的通信模式,从而选择适应当前网络环境的通信模式。当然,这部分特性的探测也是可选的。

综上,对代理服务器特性的探测大致可分为4个阶段,先后分别对应识别用户、控制数据处理、控制用户访问和应用程序通信模式特性的探测。

3.2 探测算法设计与实现

在设计代理特性探测算法时,一方面要选择对通信影响较大且容易探测的属性作为探测对象;另一方面要分析所选属性之间的内在联系从而安排探测的逻辑顺序。同时,具体属性的选取还要依据具体应用层协议的特点,包括不同协议版本的区别。

探测过程最终都落实到与代理服务器建立连接并收发数据包,因而探测算法设计的一个重要原则就是占用尽量少的本地网络、服务器等各种资源,即建立尽量少的连接数(并发的或总共的),发送尽量少的数据包,引起尽量少的数据流量,包括在代理服务器上产生尽量少的日志。

根据探测原理,本文以最为常见的HTTP代理探测为例,给出其特性探测算法流程(如图3所示)。图中4个虚线框分别对应需要探测的4个子模块。

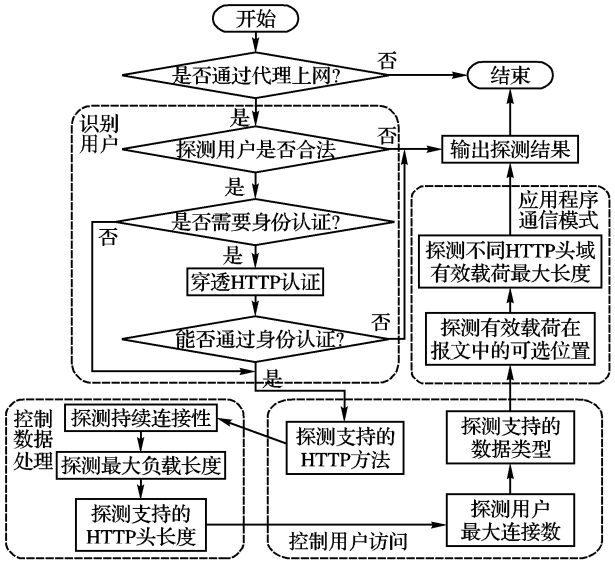

图3 HTTP代理特性探测算法流程

本文利用Socket编程实现了基于该算法思想的HTTP代理特性探测系统原型。该原型在需要访问代理服务器的用户主机上运行,主要面向Windows系列操作系统。

首先通过在注册表中查询特定键值(字符串"ProxyEnable"和"ProxyServer")来判断当前主机是否设置成通过代理访问网络。在探测代理特性时,尝试与代理服务器建立连接并向其发送若干条精心构造的HTTP请求消息[5],根据返回的应答信息来判断是否具有相关属性或具体取值。

在识别用户部分的探测中,如果代理服务器要求身份认证,通常会返回状态码为407的应答消息。这里很重要的一项工作就是穿透认证机制,使当前用户合法化,以便进行后续的探测。HTTP认证机制[6]主要有基本认证、摘要认证、集成式认证、SSL证书认证、RADIUS认证等,其中较为常见的集成式认证又涉及Kerberos V5、NTLM、Negotiate等安全协议[7]的使用。本文实现了对基于活动目录(Active Directory)[6]的Windows域环境下集成式身份认证的穿透。解决的思路是利用Windows域认证环境的单点登录(Single Sign-On,SSO)[8]机制,使用Windows系统提供的Security Support Provider Interface(SSPI)获取当前主机已有的认证信息,构造合法的带认证信息的HTTP请求消息,从而通过认证,以当前授权用户的名义进行通信。由于单点登录在Windows域环境中被普遍使用,因此这种特定认证环境下的代理探测具有一定价值。

在探测控制数据处理部分之前,先探测支持的HTTP方法。虽然该特性属于控制用户访问部分,但它直接影响到后续探测过程中数据包的构造,因而要提前进行。在控制持续连接性时,要注意使用的HTTP协议版本,只有1.1及以后的版本才支持持续连接。在探测最大负载长度时,由于不同的HTTP方法可能会对请求载荷有不同的约束(例如可以设置不允许GET方法携带载荷),要针对所支持的HTTP方法作分别处理;同时还要考虑用户预期的单数据包有效载荷长度(探测最大HTTP头长度也应考虑到这一点)。对于用户最大连接数的探测(同样需考虑用户预期的最大并发连接数),采用多线程技术实现并发的向代理服务器建立连接,各线程利用事件实现同步,并通过设置临界区实现互斥的访问共享变量。

应用程序通信模式的探测,目的是权衡用户有效数据在HTTP隧道中的位置和大小。将用户数据作为HTTP请求载荷是最普通的情况,这里没有考虑,而将数据存放在HTTP请求的开始行或消息头,实质上是利用隐蔽通道[9]进行通信。在HTTP请求消息格式中,可以携带有效数据的位置[10-11]主要有开始行的URL、消息头的Cookie、Accept等头域,其中URL和Cookie内容的随机性较大,适合直接搭载用户数据,需要对其可用性和最大长度进行探测。

4 测试与性能分析

4.1 有效性测试

在实验网络环境下,本系统能够正确探测ISA,Squid等功能强大的代理软件的多项特性,如表2所示,其中"√"表示可以配置并探测成功。

表2 代理特性探测系统有效性测试

属性	代理软件	
	Microsoft ISA Server 2006	Squid 3.0 Stable
用户合法性	√	√
身份认证	√(通过集成式认证)	未测试(见注)
持续连接性	不可配置	√
最大负载长度	√	√
最大连接数	√	√
支持的数据类型	√	√
最大HTTP头长度	√	√
最大URL长度	√	不可配置

Squid需要在编译时作配置实现基本认证功能,实现其他认证机制需要安装外部程序,较为繁琐,故未作测试。

4.2 算法性能分析

算法的意义主要体现在与不作探测的情况相比,通信数据量和通信时间有所下降。假设数据包发送速率一定,不计本地延迟和线路传输等耗时,则数据量与通信时间成正比。不妨设探测过程产生数据量为D,有效数据总量为S,单个数据包冗余数据量为R,不作探测时用户预期的单位数据包有效载荷为$P1$,探测后确定单位数据包有效载荷为$P2$。

若$P1 > P2$,则不作探测时无法正常通信,探测具有决定性作用;若$P1 = P2$,探测没有意义;若$P1 < P2$,原通信总量为$S + \frac{S}{P_1} \cdot R$,带有探测的通信总量为$D + S + \frac{S}{P_2} \cdot R$。如果算法确实提高了通信效率,则$D + S + \frac{S}{P_2} \cdot R < S + \frac{S}{P_1} \cdot R$,即

$D < \frac{S \cdot R \cdot (P_2 - P_1)}{P_1 \cdot P_2}$。这里 R 是常量,本文认为应用程序会持续工作,即 $S \to +\infty$,D 可以满足要求。

为提高算法自身的效率,除了合理安排探测的逻辑顺序以减少冗余通信外,还应在各个属性的探测中减少通信量,缩短通信时间。定性探测的流程比较简单,压缩通信量的余地不大;而定量探测的流程相对较复杂,特别是对属性最大值的探测,可通过具体的程序设计尽量减少建立连接和发送数据包的次数。这里以探测 Microsoft ISA Server 2006 的 HTTP 消息最大负载长度为例,在支持持续连接的条件下,假设用户预期有效载荷长度为 10000 字节。给代理服务器设定不同的最大负载长度值,分别以几何平均值和算术平均值逼近设定值的策略进行探测,结果如表 3 所示(误差为 10 字节),可见在同样的精度要求下,几何平均策略所得结果更接近实际设定值。连接次数和发包次数的数据对比如图 4 所示,可见几何平均策略建立连接的次数较少,而算术平均策略发包的次数较少,经权衡最终确定采用几何平均策略。

表 3　最大负载长度探测结果对比　　字节

设定值	策略		设定值	策略	
	几何平均	算术平均		几何平均	算术平均
512	512	507	6144	6143	6141
1024	1017	1024	7168	7167	7167
2048	2043	2040	8192	8192	8183
3072	3069	3065	9216	9214	9208
4096	4092	4091	10240	10000	10000
5120	5118	5117			

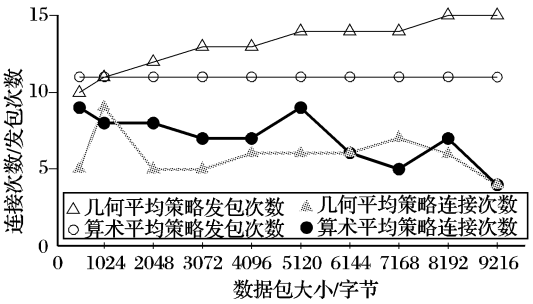

图 4　最大负载长度探测实验数据对比

5　结语

研究网络应用程序对代理服务器的适应性能有效提高通信效率,也是优化代理性能的另一个重要途径;同时,在非授权条件下,获取代理特性信息是有针对性地突破代理的基本前提。因而,研究代理服务器特性探测技术具有较为重要的意义。

参考文献:

[1] 孙青. 代理服务器安装配置与应用[M]. 北京: 冶金工业出版社, 2002: 48 - 49.

[2] 李常先, 王海. 代理服务器的安全性能指标及测试研究[J]. 计算机系统应用, 2007(5): 85 - 87.

[3] 曹元其. Squid 安全策略[J]. 网管员世界, 2006(1): 128 - 130.

[4] 邱鸿江, 楼靖华. ISA Server 在网络服务中的配置和应用[J]. 科技文献信息管理, 2003, 17(4): 7 - 11.

[5] FIELDING R, GETTYS J, MOGUL J. RFC2616, Hypertext Transfer Protocol - HTTP/1.1[S]. 1999.

[6] (美) ISEMINGER D. Microsoft Windows 2000 活动目录服务技术参考[M]. 宋书民, 陈郁红, 娄裕, 等译. 北京: 科学出版社, 2001: 117 - 118.

[7] JAGANATHAN K, ZHU L, BREZAK J. RFC4559, SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows[S]. 2006.

[8] 毛捍东, 张维明. 一个基于 Web 服务的单点登录系统[J]. 计算机工程与应用, 2004, 40(24): 18 - 20.

[9] AHSAN K, KUNDUR D. Practical data hiding in TCP/IP[C]// Proceedings of ACM Workshop on Multimedia and Security: Authentication, Secrecy, and Steganalysis. New York: ACM Press, 2002: 7 - 14.

[10] BAUER M. New covert channels in HTTP: Adding unwitting Web browsers to anonymity sets[C]// Proceedings of the ACM Workshop on Privacy in the Electronic Society. New York: ACM Press, 2003: 72 - 78.

[11] 邹昕光, 金海军, 郝克成, 等. 基于 HTTP 协议多维随机参数插入通信隐藏算法[J]. 计算机工程与应用, 2006, 42(34): 127 - 130.

(上接第 825 页)

3.3.4　实体偏序关系优先

针对形式冲突中实体间存在一定的逻辑偏序关系,可根据偏序关系对冲突实体进行冲突优先处理。可设定偏序关系中高层优先,也可设定低层优先。如对于主体冲突 p1,p2,可设定主体层次高的优先,即若 p2.s2 继承 p1.s1,则当 p1 和 p2 发生冲突时,应以 p1 为主。

4　结语

本文针对一种极具一般性的安全策略描述方法,提出了一套安全策略冲突分类定义、检测及消解方法,为基于策略的系统安全应用打下了良好的基础。

参考文献:

[1] MOORE B, ELLESSON E, STRASSNER J, *et al*. RFC3060, Policy Core Information Model[S]. USA: RFC Editor, 2001.

[2] MOORE B. RFC3460, Policy Core Information Model Extensions [S]. USA: RFC Editor, 2003.

[3] JASON J, RAFALOW L, VYNCKE E. RFC3585, IPsec Configuration Policy Information Model[S]. USA: RFC Editor, 2003.

[4] BOYLE J, COHEN R, HERZOG S, *et al*. RFC2478, The COPS (Common Open Policy Service) Protocol[S]. USA: RFC Editor, 2000

[5] LUPU E, SLOMAN M. Conflicts in policy-based distributed systems management[J]. IEEE Transaction on Software Engineering, 1999, 25(6): 854 - 869.

[6] CHARALAMBIDES M, FLEGKAS P, PAVLOU G. Dynamic policy analysis and conflict resolution for DiffServ quality of service management[C]// 10th IEEE/IFIP Network Operations and Management Symposium. Washington, DC: IEEE Computer Society, 2006: 294 - 304.

[7] CHARALAMBIDES M, FLEGKAS P, PAVLOU G. Policy conflict analysis for quality of service management[C]// Proceedings of Sixth IEEE International Workshop on Policies for Distributed Systems and Networks. Washington, DC: IEEE Computer Society, 2005: 99 - 108.