基于协议分析的 IM 阻断策略及算法分析

胡振宇,刘在强,苏璞睿,冯登国
(中国科学院软件研究所信息安全国家重点实验室,北京 100080;中国科学院研究生院,北京 100039)

摘 要: 本文根据即时通讯(IM)的协议模型,在 TCP 层和即时通讯协议层分析了阻断即时通讯的可能性,分别提出了基于 TCP 层和即时通讯协议层的协议分析的阻断策略. 在 TCP 层,通过插入 FIN 报文或 RST 报文,以关闭 TCP 连接(特别是与登录服务器的连接);在即时通讯协议层,通过对相应字段的值进行设置,使客户端进入不活动状态.

为使阻断策略能够有效地实施,本文也提出了一个消息驱动的实现算法. 通过对 Yahoo! Messenger 的阻断实验,证明了本文提出的方法能够高效地完成阻断任务(平均数据包量不超过 2.5 个). 用统计分析的方法对它们的性能进行比较,结果表明各种方法的效率在不同的背景流量下没有显著的差异,通过消息驱动算法实现的基于协议分析的阻断策略,其工作效率不受背景流量的影响. 另外,即时通讯数据的触发方式,使得发送的阻断包的数量与背景流量无关,也不对网络负载造成明显的影响. 本文的工作为实施即时通讯的阻断提供了有效的解决方案.

关键词: 即时通讯; P2P; 威胁; 协议; 阻断
中图分类号: TN911 **文献标识码**: A **文章编号**: 0372-2112 (2005) 10-1830-05

On the Policy and Algorithm to Block Instant Messaging

HU Zhen-yu, LIU Zai-qiang, SU Pu-rui, FENG Deng-guo
(*State Key Laboratory of Information Security, Institute of Software of Chinese Academy of Sciences, Beijing* 100080, *China*; *Graduate School of the Chinese Academy of Sciences, Beijing* 100039, *China*)

Abstract: Depending on the general protocol model, this paper analyzed the possibility of blocking Instant Messaging (IM) at TCP layer and IM protocol layer respectively. At TCP layer, packets of FIN or RST were inserted to close a TCP connection; at IM protocol layer, certain commands with special flag were inserted to enforce a client to be idle.

To implement these blocking policies effectively, a message-driven algorithm was offered, too. Some experiments to block Yahoo! Messenger and their statistical analyses were also presented in this paper. The results showed that all the policies implemented through the message-driven algorithm block IM effectively, their performances are independent of the back traffic, and thus, they don't impose much burden on the whole Internet. The works done in this paper provided practicable and effective solutions to block IM.

Key words: block; instant messaging (IM); peer-to-peer (P2P); protocol; threat

1 引言

作为当今非常流行的网络服务之一,即时通讯(IM)正在受到越来越多的人的青睐. 它最初的功能主要是为了给用户提供在线聊天. 随着它的应用的越来越普遍,它的功能也逐渐变得越来越强大——包括了从文字聊天、文件传输、语音和视频(聊天)、拨打电话、远程协助、邮件辅助、发送短信和浏览咨询等. 但由于缺少必要的安全措施及保障机制,它正在成为黑客攻击和病毒传播的有利环境,更不用说它能够在用户之间输送任何机密数据. 这些特性不仅为不法之徒盗窃商业机密,而且为传播反动言论提供了方便. 除了安全方面的因素外,未经授权地使用公开的即时通讯和点到点(P2P)文件共享程序还可能带来法律责任、生产力损失、带宽消耗等风险和威胁. 因此对即时通讯进行管理和控制是网络管理面临的迫切问题. 本文讨论主流即时通讯(MSN, Yahoo! Messenger, ICQ, AIM)的阻断控制问题.

目前主流的即时通讯软件大都工作在客户-服务器模式. 一般说来,客户和服务器之间有两种常用通讯方式——代理(proxy)和掮客(broker)[1].

在代理(proxy)方式下,所有的即时通讯消息都通过服务器来中转. 其优点是通讯的客户双方都主动建立与服务器的对外连接,而不是在某个端口接受外部进来的连接(企业的防火墙可能不允许这种连接). 但发送消息到服务器可能会造成延迟和隐私方面的泄漏(给服务器). 在当前的主流即时通讯中代理方式是默认的工作方式.

在掮客(broker)方式下,发往服务器的唯一的消息是与另

收稿日期:2004-12-22;修回日期:2005-06-27
基金项目:国家 863 高技术研究发展计划(No. 2003AA142150)

一个客户建立连接的请求.服务器为客户转发连接请求,帮助客户之间建立直接的连接,使用客户之间的通讯直接进行而不是通过服务器转发.例如,Alice 想发消息给 Bob,她首先发出一个请求到服务器,服务器再通知 Bob,说 Alice 想与之聊天.如果 Bob 同意,他回应服务器一个连接信息(如 IP 地址和端口号).这个连接信息就会被转发给 Alice.于是 Alice 就可直接与 Bob 建立连接,他们之间的消息将不再通过服务器而是直接到达对方.这种方式的优点是可以减轻服务器的负荷,避免隐私信息泄漏给服务器.但这种方式通常不能通过企业的防火墙(通常防火墙不允许外面发起对内的连接).

尽管存在各种安全问题,但由于其使用方便,界面友好再加上免费的诱惑,即时通讯还是被越来越多的人们使用.为了减轻即时通讯带来的威胁,人们提出了各种各样的建议[1~4].Dan Frase[4]把这些建议加以概括并指出:"最好的方针路线是建立、分发和加强与用户级别相适应的安全策略;确信病毒防护机制在服务器、网关和客户机上得以正确布署;确定适当的即时通讯解决方案;配置防火墙以阻止不可靠用户与即时通讯服务器建立连接."然而,就算采用企业级的安全方案,只要用户可以与外界相连,他就可能通过即时通讯泄漏机密信息.杜绝使用才是最安全的办法.但正如文[2]中所述,要阻止即时通讯的使用是非常困难的,简单的基于端口阻断或协议分析,甚至基于 IP 地址(或域名)阻断的防火墙并不很好地工作.因为目前大多数流行的即时通讯软件都有自动配置功能,以绕过防火墙.

本文中我们根据即时通讯的协议模型,分别在 TCP 层和即时通讯协议层分析了阻断即时通讯的可能性.提出了基于 TCP 层和即时通讯协议层的协议分析的阻断策略:在 TCP 层通过插入 FIN 和 RST 数据包来中断 TCP 连接;在即时通讯协议层通过插入状态转换命令,使客户端进入不活动状态,失去消息收发功能.

我们也提出了一个消息驱动的实现算法,并通过对 Yahoo!Messenger 的阻断实验,证明了所述方法的有效性.通过对实验结果的统计分析,发现各种方法的代价在不同的背景流量下没有显著的差异,说明通过消息驱动算法实现的基于协议分析的阻断策略的工作效率不受背景流量的影响.另外采用即时通讯数据的触发方式,使得发送的阻断包数量与背景流量无关,也不会对网络负载造成明显的影响.本文的结果为即时通讯的阻断提供了高效的解决方案.

本文中,第 2 节分析了即时通讯的协议模型,提出了基于协议分析的阻断策略;第 3 节提出并分析了消息驱动的实现算法.第 4 节通过对 Yahoo!Messenger 的阻断实验证明了它们的有效性;第 5 节进一步讨论了基于协议分析阻断策略的具体细节和适用特性.

2　即时通讯的阻断策略

根据即时通讯的体系结构,一个用户必须首先登录到服务器进行登记并获取其它用户的状态信息,然后才能与其它用户联系,进行即时通讯.在默认情况下,主流的即时通讯的服务器大都有固定的连接端口,Yahoo!Messenger 是 5050,MSN 是 1863,ICQ 和 AIM 是 5190.从这点来说,只要在防火墙上阻断这些端口,就能够切断这个即时通讯与服务器的连接,达到阻断的目的.但实际上,这些主流的即时通讯软件的客户端都可以自动配置,一旦发现默认的端口不能连接,它们自动使用其它常用端口如 Telnet(23),FTP(20/21),SMTP(25),POP(110),NNTP(119),或 HTTP(80)进行连接.并且,客户端往往会在这些应用协议上开辟一个通道,将即时消息打包成应用协议的形式,以透过端口阻断及协议分析防火墙.

主流的即时通讯软件都提供众多的域名为客户提供登录[1].例如,在当前 Yahoo!Messenger 协议中,形如 csX.msg.dcn.Yahoo!.com(其中的 X 是一个从 1 到大约 40 的数字)的域名均可使用.况且,这些域名还在不断变化,其对应的 IP 地址涵盖一定的范围,不同的域名可能映射到同一 IP 集合.即使以上的域名全都被封掉,客户端仍然可以通过代理来访问其服务器.而且这些代理的域名和 IP 地址也都处于不断变化之中,对它们一一加以封堵简直是不可能的.

目前主流的即时通讯(MSN, Yahoo!Messenger, ICQ, AIM)都工作在 TCP/IP 层上.在 TCP/IP 层之上再构建各自的即时通讯协议.如前所述,为了绕过端口阻断的防火墙,它还可能在 TCP/IP 层上加载其它应用协议(如 telnet,ftp,http 等),然后再实现自己特有的即时通讯协议.其协议栈结构如图 1 所示(虚线框中的部分专门用于透过防火墙).

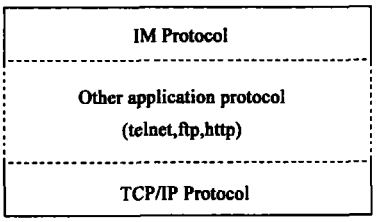

图 1　即时通讯的协议模型

对于某一具体的即时通讯来说,服务器的 IP 地址是可变的,域名是可变的,端口是可变的,就连协议甚至也是可变的,但其协议的最低层(TCP/IP)和最高层(IM protocol)是不变的.若在此两个层次实施阻断则不会受到端口或 IP 地址变化之虐.特别是其底层的 TCP/IP 协议是各种即时通讯的共同基点.在此层次实施阻断则可能抓住关键所在.

下面分别从 TCP/IP 层和即时通讯协议层来分析阻断的可能性和阻断策略.

2.1　TCP/IP 层的阻断

TCP/IP 协议是标准的网络连接与传输控制协议,大部分的网络应用程序在其之上实现互连互通.如果 TCP 连接被破坏了,其上的所有应用协议也就无法正常工作.根据 TCP/IP 的工作原理,一个 TCP 连接用一个四元组(原 IP,原端口,目的 IP,目的端口)来唯一标识.在一个 TCP 包中虽然含有一个 32 位的序号,但它仅用来确定到达的字节流的顺序.TCP 包中的校验和可以检验数据在传输过程中是否出现错误,但不能用来对数据包的起源进行认证.

建立一个 TCP 连接需要三次握手,而终止一个连接要经过 4 次握手.既然一个 TCP 连接是全双工的,因此每个方向必须单独地进行关闭.当一方要关闭一个连接时,就发送一个 FIN 给对方,进入主动关闭.当另一端收到一个 FIN,它就通知应用层,另一端已经终止了那个方向的数据输送,然后进入被

动关闭.发送 FIN 通常是应用层进行关闭的结果.

发送 FIN 来终止一个连接是一种正常的方式,也称为有序释放.除此之外,TCP 连接还可以用异常方式来终止.在 TCP 首部中的 RST 比特是用来"复位"的.一般说来,无论何时发生连接错误,TCP 都会发出一个复位报文来中途释放一个连接,而不是发送 FIN 来正常关闭.另一端一旦收到 RST 报文,就立即终止连接,并通知应用层连接复位.由于 RST 复位用于处理异常,所以无需进行半关闭的等待,也不需要经过 4 次握手,甚至连另一端的确认都不需要.因此这种关闭是即时完成的.

当前的主流即时通讯(MSN, Yahoo ! Messenger, ICQ, AIM)都工作在 TCP/IP 层上,根据 TCP 连接的安全特性以及终止关闭方法,我们可以在即时通讯的 TCP 连接上插入 FIN 报文或 RST 报文,以关闭 TCP 连接(特别是与登录服务器的连接),造成服务器不可用的假象,达到阻断即时通讯的目的.

2.2 即时通讯协议层的阻断

用户成功登录后,该用户的状态将呈现为在线状态,他可与其它客户进行即时通讯.若用户登出,则状态变成为离线,不能进行任何即时消息的收发.在客户端的运行期间,用户还可以自行改变状态,以决定对即时消息的接收或发送状态.如:联机,离开,忙碌,外出,接听电话,显示为脱机等.在 MSN 中,一个用户还可以阻止另一个特定的用户,阻止者在被阻止的客户端上将显示为脱机,该用户不能给阻止者发送任何消息.

用户的状态可分为活动和不活动两类.在活动状态下,用户可双向进行即时通讯.在不活动状态下,用户要么只能单向接收,或单向地发送某些特定消息,要么就完全不能与其他客户之间进行通讯.值得注意的是,用户在线主动将自己的状态改变为脱机,与真正的离线是有区别的.例如,在 MSN 中有两个命令使用户显示为离线状态:"CHG FLN"(离线)和"CHG HDN(隐身)".当用户用"CHG FLN"(离线)改变自己的状态时,将通知其用户列表上的其他用户,关闭所有的即时通讯.但此用户并没有真正离线,他仍然可以刷新用户列表.当用户用"CHG HDN"(隐身)改变自己的状态时,该用户在其他用户看来是处于离线状态,不能与之进行即时通讯.但该用户可以接收其它用户的状态变换通知,也能刷新用户列表.

目前的即时通讯软件都允许用户主动或被动地改变自己的状态,以接收或拒绝来自其它客户的即时消息.所谓主动改变状态,是指用户自己发出状态改变命令并通知服务器;被动改变则是指由服务器发出状态改变命令,以通知客户端(如服务器进入维护阶段,暂停服务,用户登录冲突等),迫使客户端改变状态.在 MSN 中,改变状态的命令有 OUT(离线)、NLN(上线)、CHG(改变状态)[5].在 Yahoo ! Messenger 中有服务号为 01(上线)02(离线)[6]、15(下线)和状态码 0x5a55aa56.ICQ 中则是通道 01(上线)、04(下线)[7].

对于 Yahoo ! Messenger 来说,在其 20 字节的包头中有一个二字节的服务类型字段可以用来设置服务类型号,另外有一个 4 字节的状态字段可以用来设置状态码(如图 2)[6].如果将服务类型号字段的值设置成下线、离线或其状态码字段设置为 0x5a55aa56,就可使客户端关闭消息收发活动,进入不活动状态.若包头中版本号(Version)字段为 0,则表示该命令是服务器发出的,通知客户端改变状态;否则(版本号字段不为 0),则表示该命令是客户端发出的,服务器将记录该客户端的请求,并通告其它客户端.通过伪造这些命令,和对相应字段的值进行设置,可使客户进入不能发送或接收消息的状态,达到阻断的目的.

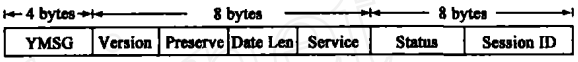

图 2 Yahoo! Messenger 的数据包头格式

3 消息驱动的实现算法

要使阻断的数据被客户端或服务器接受产生阻断效应,必须为阻断数据包设置正确的连接参数(IP 地址和端口号)和适当的序列号甚至正确的会话 ID,使之看起来是从合法的通讯方发出的.因为这些数据的动态性,必须实时地从当前的 IM 数据包中提取并进行正确推算.为此,我们提出一个消息驱动的阻断算法——Message-driven block,其阻断动作由 IM 消息激活.当调用该算法时,对它的状态进行初始化,其内容包括阻断策略的选择和阻断是否激活等.调用后,它就进入等待状态,直到被网络上的 IM 消息激活(activate).一旦被激活,该算法就根据当前的内部状态处理到来的 IM 消息,产生新的内部状态(active),在网络上输出阻断数据并调用其它算法(如报警算法).另外,它可能还要产生一些本地输出(如日志).激活完成后,该算法重新进入等待状态,直到下一次被激活.其形式化描述如下:

```
Message-driven block(block policy, inactive):
  For()
  {
    Wait for a network packet;
    If(the packet is a IM packet)
    {
      Decode the IM packet;
      Get the current TCP parameters; calculate the serial number and TrID;
      Construct a block packet according to the block policy;
      Send out the block packet;
      Invoke the alert algorithm;
      Set the current state active and log the block action;
    }
  }
```

我们来分析以上算法的性能.Message-driven block 以消息驱动的机制实现,也就是说,仅当 IM 消息到来时它才会被激活并输出阻断数据.在 IM 的空白期间(没有消息传递),阻断算法处于等待的安静阶段,不对外输出任何内容.用来阻断的数据仅仅与到来的 IM 消息相关联,其它的网络数据不会触发阻断的动作,因此也不对其阻断性能产生任何影响.另外消息驱动机制使得输出的阻断数据与 IM 消息相对应,再加上阻断数据的长度比一般的 IM 消息短得多(当在 TCP 层阻断时,其

负载长度为 0 字节),Message-driven block 算法也不会对整个网络的负载产生太大的影响.

4 不同阻断策略的实验比较

为研究第 2 节中提出的基于协议分析的阻断策略的有效性和差异性,我们针对 Yahoo!Messenger 通讯,在不同的背景流量下进行阻断实验.采用的策略分别为:发送有 FIN 标志的 TCP 的包,发送 RST 标志的 TCP 的包,发送服务号为 0x02 的包,发送服务号为 0x15 的包以及发送状态码 0x5a55aa56 的包.

实验设计:采用在第 3 节提到的消息驱动的算法来实现阻断.实验分为三组,第一组的背景流量为 1378.30B/s,第二组为 65229.30B/S,第三组为 2246.90B/S(背景流量用带参数 e 的 netstat 程序统计).三组实验除了背景流量稍稍不同外,其余条件完全相同,实验的统计量为两个,分别为从发出第一个阻断包到阻断成功所花的时间和数据包数量.阻断成功的标志是检测到真正的客户或服务器发送一个 FIN 或 RST 的 TCP 包,以关闭或终止网络连接.每组实验的每种方法的两个统计量分别提取 10 个试验值,共 300 个样本数据.

为比较不同的阻断策略方法的有效性我们采用 One-Sample Kolmogorov-Smirnov 检验,计算其均值和标准差.为比较各方法在不同的背景流量下的差异性,我们利用非参数的独立样本检验方法,采用 Kruskal Wallis 检验,分组变量是背景流量.数据用 SPSS10.0 软件计算处理.

实验条件:阻断的发起者与客户端在同一主机上,主机的配置为:CPU of P4 2.60GHz,RAM of 512MB,Windows 2000 pack 4.网卡型号为:Intel(R) PRO/1000 MT Network Connection.

实验数据的统计结果分别如表 1 和表 2 所示.

表 1　五种策略方法所花费代价的均值和标准差

阻断策略	FIN		RST		SV02		SV15		ST	
	time	count	time	count	time	count	time	count	time	count
样本	30	30	30	30	30	30	30	30	30	30
均值	356.87	1.50	184.83	2.33	371.40	1.60	341.20	2.10	319.83	2.00
标准差	234.20	.63	245.00	.48	226.79	.56	243.20	.61	245.35	.59

表 2　各策略方法在不同的背景流量下的差异性比较

(Kruskal Wallis Test,Grouping Variable:back_traffic)

阻断策略	FIN		RST		SV02		SV15		ST	
	time	count	time	count	time	count	time	count	time	count
χ^2	5.588	.205	4.609	2.030	.347	1.576	1.931	.530	.809	.580
自由度	2	2	2	2	2	2	2	2	2	2
P 值	.061	.903	.100	.362	.841	.455	.381	.767	.667	.748

*在表一和表二中,FIN 表示发送有 FIN 标志的 TCP 的包,RST 表示发送 RST 标志的 TCP 的包,SV02 表示发送服务号 0x02 的包,SV15 表示发送服务号 0x15 的包,ST 表示发送状态码 0x5a55aa56 的包.time 表示所花时间(毫秒),count 表示包的数量.

5 结论及分析

从表一的结果可以看出,每种策略方法成功阻断 Yahoo!Messenger 通讯的平均数据包数量都不超过 2.5 个.说明基于协议分析的各种阻断策略都能以极高效的效率实施阻断.

从表二对各种策略方法在不同的背景流量下的差异性比较来看,其显著性概率 P 值均大于 0.05(Asymp.sig(2-tailed)).从时间差和数据包量来说,背景流量对各种阻断策略的效率没有显著的影响.

从消耗时间的实验结果来看,各种方法的之间的标准差较大(上百毫秒).这主要是因为用来阻断的数据包与客户端或服务器发出的正常数据包之间存在竞争关系.如果用来阻断的模拟数据包先于被模拟的真实数据包到达接收方,阻断就会起作用.否则,用来阻断的模拟数据包就会被接收方作为重复包丢弃.

在所有的阻断策略中,最为直接的是 RST 方法.只要一收到 RST 包,客户器或服务器立即中断连接,从而使即时消息也无法收发.如果伪造的 RST 数据包是发往客户端的,则收到该 RST 包的客户端立即终止消息的收发,而如果伪造的 RST 数据包是发往服务器,则服务器先终止连接,而此时客户端并不知道,它将继续向服务器发送消息.服务器收到客户端的消息后再回答一个 RST 包(因为它已经终止连接),这时客户端才发现连接中断而停止收发消息.

FIN 方法也可以快速引起关闭连接.但是,由于是插入的 FIN 包,它将导致客户端和服务器都进入被动关闭的状态,使双方同时等待对方的关闭应答,直到超过时限而自动放弃.从收到 FIN 包到完全关闭,通常要等待比较长的一段时间(这与我们的实验结果不矛盾,我们的实验是以检测到一个 FIN 包为标志,虽然没有完全关闭,但已不能进行数据通讯).

SV02,SV15 和 ST 方法工作在 Yahoo!Messenger 协议层,并且只能从单个方向起作用.即,只有作为发向服务器的客户端数据包,这种方法才会起作用.相比之下,工作在 TCP 层的 FIN 和 RST 方法则可在两个方向上发挥作用.

直觉上,SV02,SV15 和 ST 方法应该比 RST 和 FIN 方法要优雅,但实际观察的结果表明它们之间没有明显的差别,所有的五种方式都可阻断 Yahoo!Messenger.被阻断后的客户端表现为与服务器失去连接.

实验结果不仅表明了基于协议分析的阻断策略是高效的,也表明了其效率不受背景流量的影响.这是因为,我们采用的机制是:以即时通讯数据来触发阻断数据.因此,阻断数据包仅与即时通讯的流量相关,而与背景流量无关.另外,由于采用以即时通讯数据来触发阻断数据的机制,使我们的阻断对网络负载不会造成明显的影响.

6 进一步研究的方向

本文针对主流即时通讯的阻断进行研究.除了主流的即时通讯系统外,当前还有相当多的其他即时通讯系统也在网络上运行.如何对其他非主流即时通讯系统,特别是实现了加密通讯的系统进行阻断,是下一步工作的一个方向.另外,本文的研究是基于目前的网络协议(IPV4)进行的,随着下一代

网络协议(IPV6)的推广,网络通讯在安全性和可靠性方面可能会有所加强.在下一代网络协议(IPV6)环境中如何对即时通讯的进行阻断控制,是另一个要研究的方向.

参考文献:

[1] Neal Hindocha and Eric Chien. Malicious Threats and Vulnerabilities in Instant Messaging[OL]. Symantec Security Response WHITE PAPER, http://securityresponse.symantec.com/avcenter/reference/malicious.threats.instant.messaging.pdf.

[2] Neal Hindocha. Threats to Instant Messaging[OL]. Symantec Security Response WHITE PAPER, http://securityresponse.symantec.com/avcenter/reference/threats.to.instant.messaging.pdf.

[3] Symantec Corporation. Securing Instant Messaging[OL]. Symantec white paper, March 6, 2003. http://securityresponse.symantec.com/avcenter/reference/secure.instant.messaging.pdf.

[4] Dan Frase. The Instant Messaging Menace: Security Problems in the Enterprise and Some Solutions[OL]. http://www.sans.org/rr/whitepapers/threats/479.php.

[5] R Movva, W Lai. MSN Messenger Service 1.0 Protocol[OL]. August, Microsoft, 1999. http://www.hypothetic.org/docs/msn/ietf_draft.php.

[6] Venkat, Yahoo Messenger Protocol (ver 10)[EB/OL]. 2003, http://www.venkydude.com/articles/yahoo.htm.

[7] Alexandr Shutko, OSCAR(ICQ v7/v8/v9) protocol documentation[EB/OL]. 2005, http://iserverd1.khstu.ru/oscar/families.html.

作者简介:

胡振宇 男,1968 年 8 月出生于河南汝州,博士,主要研究领域为信息安全. E-mail: Hu_zhyu@sina.com.cn.