第 33 卷　第 16 期　　计　算　机　工　程　　2007 年 8 月
Vol.33　No.16　　Computer Engineering　　August 2007
·安全技术·　文章编号：1000—3428(2007)16—0111—03　文献标识码：A　中图分类号：TP393.08

Link-padding 流量伪装原型的设计与实现

胡柏炅，李　明
(华东师范大学信息科技学院，上海 200062)

摘　要：在计算机通信网中，窃听者能够从通信序列的统计特征中获取信息，发动流量分析攻击，由此造成链路情况的机密泄漏。该文提出了基于 IP 的以太网 Link-padding 流量伪装方法并设计出其原型，借助 Libnet，Libpcap 和 OpenSSL 的 C 语言库，在 Linux 系统上已对其进行了基本实现。

关键词：Link-padding; 流量分析；流量伪装

Design and Implementation of Link-padding Traffic Camouflaging Prototype

HU Bai-jiong, LI Ming
(School of Information Science and Technology, East China Normal University, Shanghai 200062)

【Abstract】In computer communication networks, traffic series contain statistical patterns that are useful for eavesdropper to collect information for launching traffic analysis attacks. This paper presents a system scheme and a prototype to prevent traffic analysis in IP-based Ethernet based on link padding methodology. The prototype is built on a Linux machine by using three C language libraries, namely, Libnet, Libpcap and OpenSSL.

【Key words】 Link-padding; traffic analysis; traffic camouflaging

攻击者通过在一台主机或一个网络附近进行监听，观测进出这个通信点的流量情况。通过对平均值、方差或熵等流量特征值采样并进行统计特征识别以后，关于一条信道的行动或意图等信息便可被分析出来。因此，无论传递的数据如何被有效的加密算法保护，仅仅通过对流量本身的观察和分析，攻击者也能进行信息的搜集，这对于通信机密要求很高的网络应用，例如军事及商业等，是不可忽视的安全威胁。传统意义上的数据保密、认证和完整性保护等网络安全措施已经不足以全面保证通信的安全，流量伪装技术由此应运而生。Link-padding 正是抵抗流量分析攻击的常用伪装手段。

1　Link-padding 流量伪装原理及网络模型

仙农保密原理提供了 Link-padding 流量伪装的理论依据：若能变化原始通信流量，使其符合预定的统计特征模式，信道中的流量将失去分析意义。Baran 在此基础上提出了通过在原始数据流中填充伪数据包的方法来实现隐藏两个通信主机间真实流量的目的。通过填补使得数据包长度一致，并插入伪数据包完全改变流量的统计特征，该方法在网络通信中被称为 Link-padding。

本文的 Link-padding 原型框架参考了文献[1]中提出的流量正态化理论模型，但并未考虑匿名通信，也未采取正态化伪装流量模型。

Link-padding 原型所应用的环境为文献[3,5]中提出的网络模型，即假设一个基于 IP 的以太通信网由两个受保护的子网及一条未受保护的信道组成(图 1)，子网内的所有通信对于网外监听者不可见，而信道可能是公共的或易于搭线监听的可供流量分析的传输媒介，例如因特网。为避免子网 A 与子网 B 中主机之间通信的流量在不安全信道中被截获分析，布置在子网边缘的网关 1 与网关 2 将完成双向流量的 Link-padding 伪装。两个网关将各自运行两个进程：伪装进程负责向原始数据流填补和插入伪数据；接收进程负责从伪装过的流量中丢弃伪数据以恢复原始通信。两个进程功能相反，故本文只针对伪装进程进行讨论。

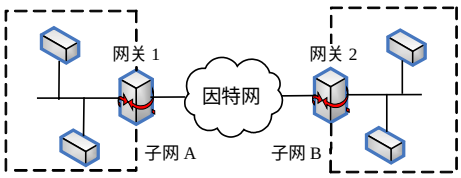

图 1　Link-padding 原型所应用的网络模型

2　Link-padding 流量伪装原型设计

2.1　流量伪装

在文献[1]中，原始流量伪装后的数据流应该由发送间隔相同且长度相同的分组构成。然而文献[3～5]却在理论上证明：无论攻击者对何种流量特征进行采样分析，发送间隔不断变化比发送间隔等长的伪装方式更能有效抵御流量分析攻击。因此，本文采用等长数据分组及发送速率不等的 Link-padding 方法。

当原始数据流中的数据分组长度不够时，补充数据将被填补在其尾部以使其达到预先设定的统一长度，如果这一长度被定义为 MTU，对大多短数据包附加过多无用数据会造成链路带宽的严重损耗，方案定义每个包长度为 500B，原始长度超出者将被分片处理再进行填补。当原始流量不满足预先

基金项目：国家自然科学基金资助项目(60573125)
作者简介：胡柏炅(1983－)，男，硕士研究生，主研方向：网络安全；李　明，教授、博士生导师
收稿日期：2006-08-22　**E-mail：**hubjune@gmail.com

要求时，等长伪数据分组将被插入其中以达到预期的包速率。所有的伪数据内容及发送间隔数都是随机生成的，随机数来源于 Linux 系统下的设备/dev/urandom，所产生的随机序列不符合任何特定分布，这使得监听者无法获知当前的流量是否经过伪装。经过处理的数据包，其 MAC 目的地址都将被替换成以太网广播地址 ff-ff-ff-ff-ff-ff 以使其能成功地被上游路由器接收并转发。

2.2 协议层伪装及其他问题

以下分 4 部分讨论如何伪装与伪造分组，以使附加的伪装流量能正确地被接收进程识别并丢弃，而攻击者却无法从中分辨用以恢复真实的通信流量。根据设计所涉及的数据域，分片后的单个 IP 分组可被表示成<cksum, len, payload>，其中，cksum 位于传输层(如 UDP 与 TCP 层)，是除 IP 首部外整个分组的校验和，仅出现在分片后的第一片 IP 分组中；len 为自定义的 16bit 字段，位于载荷的最前部，指示分组的真实长度；payload 是数据包中除协议层之外的其他载荷。

2.2.1 总长度字段伪装

每一个填补后的分组的 IP 层中用以指示整个 IP 分组长度的总长度字段皆为 500B，如此，窃听者与接收进程都无从知晓这些等长分组中哪些接受过填充，哪些是纯伪数据包。因此，每个报文中必须存在指示真实分组长度的数据域，IP 选项域经常会被用来记录路由，源路选路等，不能被自定义使用，故在每个分组的载荷域前插入了一个 16bit 的字段 len 以指示真实的分组长度，并且这个字段必须经过加密保护。这意味着，原始报文应按照 498B 的长度进行分片。

2.2.2 传输层校验和伪装

部分 IP 报文分片后其每一片分组长度正好为 500B，不需要填充伪数据，cksum 的计算覆盖整个报文，而对于大多需要填充的分组来说，校验和仅覆盖原始的数据部分。组装报文后通过计算 IP 包的校验和并与 cksum 比较，攻击者便可分辨出哪个数据包被填充过。为了避免这类分析，无论何类报文，cksum 字段都需要加密。

2.2.3 组合加密

密钥的协商将通过 SSL 协议来完成，这里用 k 来代表密钥，每隔一段时间 k 将被更新一次以保证其新鲜度。传输层首部仅出现于未分片或分片后的第一段 IP 分组中，这类 IP 包可表示为<cksum, len, payload>，经过伪装之后成为<{cksum}k, {len}k, payload+(pad)>，其中，cksum 和 len 都需要被加密。这两个域长度都为 16bit，然而本文采用的 AES 的 CBC 模式加密算法要求输入数据模块的大小至少为 256bit，因此，载荷的前 224bit 数据被用来和这两个域组合作为加密算法的输入，加密后的数据同样依次被分为 3 份分别放入相应的区域。这样，加密后数据包的形式为< {cksum, len, payload'}k, payload''>，其中，payload'和 payload''分别为填补后载荷的前 224bit 内容和其余部分。分片后非首段分组并不存在传输层首部，可被表示为<len, payload>，经过伪装之后成为<{len}k, payload+(pad)>，所需加密的仅包含一个域，故载荷前 240bit 数据需要和 len 组合，一同参与加密运算，最终形式为<{len, payload'}k, payload''>，其中，payload'和 payload''分别为填补后载荷的前 240bit 内容和其余部分。

2.2.4 伪数据包的插入

伪数据包是为了产生特定流量而伪造的毫无意义的 IP 分组，与经过填补的分组不同，伪数据报文在组装后整个都需要被接收进程丢弃。首先，随机选取 40B～1 500B 作为报文长度，选取 TCP 及 UDP 作为传输层协议，据此来伪造 IP 报文，接着对 cksum 最高位取非，这样接收进程在重新组装，解密之后能通过校验和计算判断其为伪 IP 报文，做丢弃处理。分片后再进行插入 len 字段，填补等处理，详细过程不再赘述。为了保证分组能够从源子网正确路由到目的子网，IP 源与目的地址将分别从两个集合中随机抽取，这两个集合应包含预先获取的属于通信双方子网的所有主机 IP 地址。

2.3 恢复原始通信

接收端首先对每一个分组的 len 及 cksum(若有)字段解密，根据 len 判断真实分组长度去除附加在分组尾部的伪数据。根据 IP 首部所提供的分片信息，接收进程对分组进行组装恢复完整的 IP 报文，再进行校验和计算，与 cksum 进行比对，若完全错误则判断为传输差错要求源端重发，若仅错最高位则视其为伪造报文予以丢弃，其余的报文则被转发去子网，这样便恢复了原始的通信数据。

3 Link-padding 流量伪装原型的实现

3.1 程序框架及算法

原型的程序框架被分为帧缓冲，包捕获及填补过程，伪数据包构造及包发送过程 3 个部分。

(1)帧缓冲。帧缓冲是为解决输入与输出的数据速率不平衡造成拥塞而引入的，已经伪装完成而未被发送的分组被包装成以太网数据帧后暂存在这里。缓冲区遵循先入先出原则，每个单元长为 514B，相当于 IP 分片的 500B 加上以太网首部的长度，缓冲区被设计为滑动方式以使其可被重复利用。

(2)包捕获及填补过程。一旦在网络入口端有数据帧被捕获，该过程便被调用来完成分片，数据包填补及协议层伪装，将处理后的分组存入帧缓冲，具体流程如图 2 所示。

(3)伪数据包构造与包发送过程。该过程受到随机定时器的调度，定时中断触发后，此过程将根据帧缓冲情况伪造 IP 分组或从帧缓冲区内取出一帧数据从网络出口端向上级路由器发送，具体流程如图 3 所示。

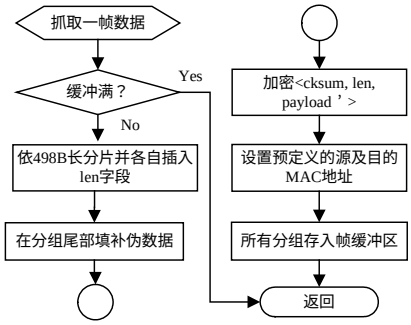

图 2 包捕获及填补流程

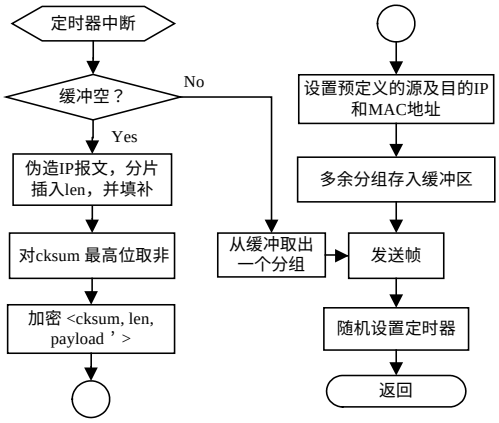

图 3 伪数据包构造与包发送流程

3.2 Libnet, Libpcap 与 OpenSSL 的应用

程序需要实现网络包的捕获，IP 报文的分片、构造及发送，公共信道上的密钥协商及加密等基本功能，本文借助 3 个开源的 C 语言库在 Linux 上完成了软件的基本开发。其中，Libnet 提供的接口函数实现和封装了数据包的构造和发送功能；Libpcap 提供了网络包捕捉的 API 函数；OpenSSL 完整实现了 SSL 及 TLS 协议，简化了在完成密钥协商及 AES 的 CBC 模式加密功能时的编程。

3.3 实验结果

图 4 显示了在一个平均字节速率在 10KB/s 左右的正常原始流量与进行 Link-padding 伪装后的流量对比。由于各个分组的发送间隔随机，单位时间内伪装后的通信维持着约 25KB/s 的速率。同时，由于若个时间段内包发送间隔较小而不断发生突发高速率流量，因此形成图中的尖脉冲。与原来的数据流相比，伪装后的流量统计特征分布发生了大的改变，不仅与前者毫无关系，而且不符合特定随机分布，对于攻击者已失去分析意义，因此，Link-padding 的原型是具有实际应用价值的。

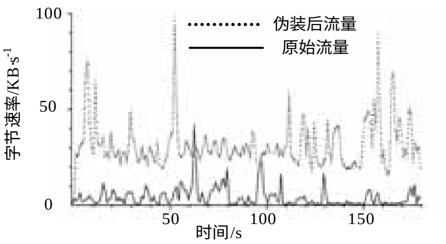

图 4 伪装前后的流量对比

4 结束语

本文基于 IP 以太网中子网间通过公共信道通信的网络应用环境，提出了一个基于 Link-padding 方法的流量伪装原型。该原型的设计不仅对流量的统计特征进行了变化，还实施了协议层的伪装，使窃听者无法从已伪装的流量中重新恢复出原始通信流。该理论原型在 Linux 系统上得到了初步的实现，实验结果证明，Link-padding 流量伪装机制在实际网络环境中是可行的。

参考文献

1 Liu D X, Chi C H, Li M. Normalizing Traffic Pattern Anonymity for Mission Critical Applications[C]//Proc. of the 37th Annual Simulation Symposium, Arlington, VA, USA. 2004.

2 Guan Y, Xuan D, Shernoy P U, et al. Camouflaging Network Traffic for QoS-Guaranteed Mission Critical Applications[J]. IEEE Trans. on Systems, Man, and Cybernetics, 2001, 31(4): 253-265.

3 Fu X W, Graham B, Riccardo B, et al. Analytical and Empirical Analysis of Countermeasures to Traffic Analysis Attacks[C]//Proc. of the 2003 International Conference on Parallel Processing, Kaohsiung, Taiwan, China. 2003.

4 Fu X W, Graham B, Riccardo B, et al. On Countermeasures to Traffic Analysis Attacks[C]//Proceedings of the 2003 IEEE Workshop on Information Assurance. United States Military Academy, West Point, New York. 2003.

5 Fu X W, Graham B, Riccardo B, et al. On Effectiveness of Link Padding for Statistical Traffic Analysis Attacksp[C]//Proc. of the 23rd IEEE International Conference on Distributed Computing Systems, Providence, Rhode Island, USA. 2003.

(上接第 110 页)

由于[2,p-1]是乘法群，则可以定义游戏G_{n1}等价于G_n，其中，G_{n1}为$<a, b, msg * c>$，且G_{n1}的概率分布与G_n的相同。如果被动上下文族C_n[]可以赢得这个游戏，则可以证明安 全性。

设 $v \in$ Obs，有 $\exists n_0$，$\forall n \geq n_0$，| $\mathrm{Prob}[C_n[P_n] \overset{eval}{\Rightarrow} v]$-$\mathrm{Prob}[C_n[Q_n] \overset{eval}{\Rightarrow} v]$ | > 1/poly(n)

设$<a, b, c>$和$<d, e, f>$为用户A在G_{n1}中生成的 2 张牌，设进程R(p, g, a, b, c)为

$$\overline{public}<p> \mid \overline{public}<g> \mid \overline{AB_1}<a> \mid BA(x).\overline{AB_2}<c> \mid AB_1(y).\overline{BA}<b>$$

对于$<a, b, c>$和$<d, e, f>$，R(p, g, a, b, c)和R(p, g, d, e, f)中的一个与P相似，另一个与Q相似。假定$\mathrm{Prob}[C_n[P_n] \overset{eval}{\Rightarrow} v] > \mathrm{Prob}[C_n[Q_n] \overset{eval}{\Rightarrow} v]$，相反的情形是对称的。

计算R(p, g, a, b, c)和R(p, g, d, e, f)来确定哪个三元组为$< g^u, g^v, g^{uv} >$，如果 v 在 2 次计算中都出现或都不出现，则得不到任何有用的信息。由此抛掷硬币，以相等的概率在$<a, b, c>$和$<d, e, f>$中进行随机选取，可以确定在产生 v 的这 2 个进程中使用了该三元组的进程就是$< g^u, g^v, g^{uv} >$。由于G_{n1}中牌的概率分布与P_n和Q_n生成的三元组的概率分布相同，得到正确三元组的概率为 1/2$poly(n)$，与假设矛盾，因此P_n和Q_n是不可区分的。证毕。

4 结束语

本文基于计算复杂性理论，提出了一个分析概率函数与安全协议的形式化方法，该方法是 Spi 演算的改进，其定义了概率可观察等价性，并用概率可观察等价性表示协议的安全性质；将密码学运算进行概率多项式时间的处理，使用进程代数来定义概率多项式时间进程。最后分析和证明了一个协议的安全性质。

参考文献

1 Berezin S. Model Checking and Theorem Proving: A Unified Framework[D]. Carnegie Mellon University, 2002.

2 Syverson P. Towards a Strand Semantics for Authentication Logic[Z]. (2004-07-15). http://chacs.nrl.navy.mil/publications/CHACS/1999/1999syverson-svo.pdf

3 Shmatikov V. Probabilistic Analysis of Anonymity[C]//Proceedings of the 15th IEEE Computer Security Foundations Workshop. 2002.

4 Abadi M, Jurgens J. Formal Eavesdropping and Its Computational Interpretation[C]//Proceedings of the 4th International Symposium on the Theoretical Aspects of Computer Software. 2001: 82-94.

5 Abadi M, Gordon A. A Calculus for Cryptography Protocols: the Spi Calculus[J]. International Journal on Information and Computation, 1999, 148(1): 1-70.

6 Glabbeek R, Smolka S, Steffen B. Reactive, Generative and Stratified Models of Probabilistic Processes[J]. International Journal on Information and Computation, 1995, 121(1): 59-80.