网络应用

文章编号: 1000-8608(2005)S-S237-05

背景流量中报文负载的字典构造方法

龚 俭[1,2], 吴 雄[1,2], 杨 望[1,2]

(1. 东南大学 计算机科学与工程系, 江苏 南京 210096;
2. 江苏省计算机网络技术重点实验室, 江苏 南京 210096)

摘要: 背景流量的构造是决定网络测试质量的重要方面. 在目前背景流量的生成中, 报文的负载一般采用随机串进行全部或部分填充. 该方法容易引入误报, 造成测试结果的不准确, 所构造的流量中报文对被测系统的压力不可控制, 在实时测试中随机串的生成降低了测试系统的性能, 而在离线测试中为存储负载中的随机串需要庞大的空间. 针对这种方法的不足, 设计了一种字典填充方法, 报文的负载从字典中选取, 而字典的内容从被测设备所能监控的网络事件的特征中提取. 该方法可以减少随机串方法所带来的问题, 实验证明是有效的.

关键词: 背景流量; 报文负载; 字典; 网络测试
中图分类号: TP393 **文献标识码**: A

0 引 言

网络流量检测设备的测试主要分为功能测试和性能测试. 功能测试通过播放网络设备所能检测的网络事件流来反映该设备所具功能的完备性. 性能测试是在模拟不同的实际环境下, 检测网络设备的承受强度. 性能测试一般的方法为截取或模拟背景流量; 通过与特定网络事件流混合播放, 来检测网络设备的各项指标.

目前在构造背景流量的方法中, 普遍采用软件平台模拟实际的网络流量. 而这种模拟的网络流量, 报文的负载一般采用随机生成的字符串全部或部分填充. 这种方法在测试某些以报文的负载为观察对象的网络流量检测设备(如 NIDS、应用层防火墙等)时, 存在以下几个问题[1]: (1) 随机生成的字符串有可能与网络事件特征冲突而引入误报, 造成测试结果的不准确; (2) 因为随机字符串的内容无法控制, 造成所构造的流量中报文对被测系统的压力不可控制; (3) 在实时测试中, 因为随机字符串生成效率的影响降低了测试系统的性能, 而在离线测试中为存储负载中的随机字符串需要庞大的空间. 针对这种方法的不足, 本文提出了报文负载字典构造的方法. 报文的负载从字典中选取, 而字典的内容由不会引入误报的正常字符串和能调节背景流量压力的特征串变型组成. 该方法克服了随机串填充的缺点, 能有效控制背景流量中报文的负载对测试设备的压力, 实验证明是有效的.

1 现有背景流量构造方法

在已有的测试中, 背景流量的构造方法各不相同. 通过对已有的构造方法研究, 总结为以下 3 种情况:

(1) 用类似于 SmartBits 的硬件流量发生器产生背景流量[4]

这种方法普遍应用于防火墙和路由器等网络设备的测试中. 使用 SmartBits 等硬件设备虽然可以产生足够大的背景流量, 但是无法使流量特征符合真实的流量特征, 最多只能使背景流量的统计特性和实际流量的统计特性相同, 例如使流量大小和平均包长相等. 硬件设备模拟的流量无法反映实际流量的协议组成、峰值、波动变化等特征. 所以这种构

收稿日期: 2005-08-06.
基金项目: 国家 973 计划课题资助项目(2003CB314803); 江苏省网络与信息安全重点实验室资助项目(BM2003201).
作者简介: 龚 俭(1957-), 教授, 博士生导师, 研究方向: 网络安全, 网络行为学; 吴 雄(1981-), 男, 硕士生, 研究方向: 入侵检测系统的测试与评估, E-mail: xwu@njnet.edu.cn; 杨 望(1979-), 男, 博士生, 研究方向: 入侵检测系统的测试与评估, E-mail: ywang@njnet.edu.cn.

造方法不适用于对 NIDS 一类的网络设备进行测试.

(2) 使用真实的网络流量或系统日志[2]

这种测试方法使用从实际运行环境中采集的网络流量或系统日志作为测试的背景活动. 这种测试方法能够正确反映被测系统在真实工作环境中的检测能力. 但是存在以下一些问题:

①对于实际网络特别是主干网络的数据采集和存储需要很高的采集能力和巨大的数据存储空间. 受到存储空间的影响, 预先采集好的数据也只能作为较短时间测试的背景流量.

②背景流量中未知的特定网络事件将对网络设备的检测结果造成干扰. 因此很难通过这种测试方法确定被测系统的误报率.

③由于真实背景流量中牵涉到的隐私问题, 测试使用的数据集合和测试的结果不适合公开发布.

由于以上问题的存在, 使用真实的网络流量或系统日志的方法极少被用于实际测试活动.

(3) 模拟实际的网络环境, 通过测试平台人工构造背景流量[3,4]

使用测试平台来生成测试数据是目前最为通用的测试方法. 该方法生成的背景流量中, 报文的负载除了高层协议字段固定外, 所携带的内容是由随机串组成. MIT 的林肯实验室在测试 NIDS 时采用了这种方式, 精心构造的测试数据模拟了一个美军空军基地的日常网络流量. 这种方法的好处在于可以确认背景流量中不会包含任何特定网络事件数据. 测试的数据集可以被用于重复测试. 由于不牵涉到隐私数据, 测试的数据集也可以公开发布. 缺点在于构造一个测试平台的花费和工作量都很大. 虽然通过测试平台人工构造背景流量也存在一些问题, 但是通过和其他两种方法的比较, 这种方法在实现难度和测试效果上是最可行和最实用的.

2 背景流量中报文负载随机生成的不足

与测试其他网络设备(如防火墙)不同, 在测试 NIDS 这类需要考查报文负载的网络设备时, 背景流量中报文负载的内容很重要. 从第 1 章的分析中可以看到, 除了使用真实流量外, 其他的方法中报文的报文头部的分析, 还会在报文的负载中检测是否有相应的网络事件特征字符串. 而随机生成的报文负载有可能与网络事件特征串冲突(因为随机生成, 有可能产生与网络事件特征串相同的字符串), 引发误报. 对冲突的概率进行了分析. 假设字符串随机生成概率模型采用均匀分布模型, 样本空间是 256(计算机系统用一个字节存储字符), 背景流量中报文的长度为 L, 网络事件特征字符串的长度为 l, 在报文的负载中出现特征串的概率为 P. 经过分析可得

$$P = 1 - \left| 1 - \left| \frac{1}{256} \right|^{l} \right|^{L} \qquad (1)$$

表 1 攻击特征串长度与冲突概率的关系 ($L = 800$)

Tab. 1 The relationship between length of signature and probability of conflict ($L = 800$)

特征串长度	冲突概率
1	0.956 63
2	0.012 13
3	0.000 047 68
4	0.000 000 186 26

从表 1 的数据可以发现, 当特征字符串的长度小于等于 2 时, 冲突的概率很大. 当特征字符串的长度大于 2 时, 一个报文产生冲突可能性比较小, 但在构造背景流量时一般会有几百万个报文, 在这些报文中冲突发生的可能性就比较大了. 对目前最著名最活跃的开放源码 NIDS 项目 snort 的检测规则进行了统计分析.

由表 2 可知 snort 的规则中, 特征串小于等于 4 的规则数有 974 条. 从 snort 的规则分析中可以看出, 目前很多网络事件的特征都是由比较短的特征串描述. 如果用随机生成报文负载的背景流量来进行测试, 将会产生一定的误报, 对测试结果的正确性造成比较大的影响.

表 2 Snort 规则分析

Tab. 2 Analysis of snort signature

攻击特征串长度	规则数
1	230
2	126
3	55
4	563

两部分,报文头部检查和报文负载的字符串匹配所耗费的时间. Antonatos 等于 2004 年的研究表明[5],NIDS 在实际环境中处理负载上所耗费的时间是所有处理时间的 40% ~ 70%,并且用随机串填充的报文与实际的报文相比,对 NIDS 的压力偏小. 由此可见,背景流量中报文负载的内容是影响网络设备性能的一个重要方面. 传统的报文负载随机字符串填充的方法,虽然可以控制头部信息,但因为负载部分随机生成,无法控制负载的内容,当然更无法控制负载对设备的性能所造成的影响. 因此使用随机填充的背景流量无法全面评估所需测试的网络设备.

(3) 对测试系统的影响

实时测试系统的性能很大程度上取决于报文生成的速度. 如果报文的负载采用随机串填充,将严重影响测试系统的性能. 在计算机系统中无法产生真实的随机数,因此一般利用随机数发生器来模拟随机数的产生. 但不管采用何种伪随机数发生器,效率都非常低,使报文填充过程变得很慢,影响实时测试系统的性能.

在离线测试系统中,流量在播放之前事先生成,因此报文负载随机填充不会影响系统的性能,但为了存储这些报文的负载,却需要庞大的存储空间. 当产生数据的流量为 100 Mbps 时,进行 1 h 连续测试所需的总数据量为 $100 \times 60 \times 60/8 = 45\ 000$ MB = 45 GB. 由此可见如果进行连续测试,数据将因为没有足够的存储空间而无法进行存储. 即使采取某些压缩方法,收效也甚微.

3　报文负载字典填充方法

本章首先讨论了设计的原则,然后给出了新方法实现的框架,最后是实验与评估.

3.1　新方法设计原则

为了解决随机字符串与网络事件特征串可能冲突的问题,一个简单的方法就是用固定的字符串来代替随机生成的字符串. 该字符串与被测系统使用的特征不存在冲突,使用该固定字符串填充所有报文的负载. 但该方法也存在着两方面的问题: ①固定字符串给被测系统提供了信息;被测系统可以从报文中出现该字符串来判断报文不是特定网络事件报文,并根据这一点提高自己的检测率和降低误报率. ②固定字符串的内容有一定偶然性,可能对不同测试系统的影响有差异,使得测试缺乏公平性.

针对随机字符串填充方法和固定字符串填充方法的不足,作者设计了报文负载字典构造方法. 在填充报文的时候并不选用固定的某一个字符串,而是预先生成一定数量的字符串,并验证这些字符串与被测系统的特征没有冲突. 用这些固定的字符串组成字典,在填充报文负载的时候,随机选择字典中的某一项进行填充. 这种方法不但避免了产生误报,而且因为选择具有随机性,可以防止被测系统作弊,保证了测试的公平性.

这种方法在提高测试系统性能方面也有很大的作用. 在填充负载的时候,字典文件可以先载入内存,因此字符串在内存中的复制取代了随机串的生成. 作者对这两种方法进行了实验比较,实验主机为双 Xeron2.4 GB 处理器,1 GB 内存,操作系统为 RedHat9. 随机数由 C 标准库提供的 rand 函数产生. 实验表明,内存中复制 500 字节数据的时间为 7 μs,而用随机串填充 500 字节数据的时间为 2 772 μs. 由此可见使用字典法填充将极大提高测试系统的性能. 而在存储报文的时候,不需要存储整个负载,而只要存储二元组(index, length),以标识所需填充的负载在字典中的位置,每个报文的负载只需 8 字节的存储空间,这也将在很大程度上减少存储的压力.

这种报文负载字典构造方法虽然解决了冲突问题和测试系统的性能问题,但仍然存在负载压力不可控的问题. 针对这种情况,作者作了进一步的改进. 网络设备在检测报文负载中是否存在某个特征字符串时,用到了各种字符串匹配算法. 目前常用的字符串匹配算法(如 B-M、KMP 等),在处理相同长度但内容不同的字符串时,进行字符比较的次数是不同的. 如果字符串中含有所要查找的特征串,那么算法进行字符比较的次数将多于处理不包含特征的字符串. 通过分析这些算法,可以构造出这些特征串的变型,相应的匹配算法处理这些特征串变型时与处理特征串所需进行的字符比较次数相同,但检查的结果是匹配失败.

本文设计的变型的具体方法是: 找出算法在检查字符串时最后进行匹配的位置,对于特征串相应的位置用其他字符替代后得到特征串的变型. 以 B-M 算法为例,B-M 算法是从后往前作匹配,因此字符串的第一位也就是最后匹配的位置. 假设某特征串为 $(a_1, a_2, \cdots, a_n)$,算法需要耗费 n 次字符来判定该字符串. 选取 $(b, a_2, a_3, \cdots, a_n)$ 为该串在 Boyer-Moore 算法下的变型. B-M 算法同样需要耗费 n 次字符比较的时间才能判断该字符串不是特征

串.

在实际应用中,根据测试需求,首先生成背景流量的报文头部,并按报文发送顺序将报头组成待填充报文头部队列. 依次从队列中取出报文头部,根据字典填充方法生成负载与报头拼接后形成完整的报文,最后将该报文直接发送至网络或者按顺序存储. 字典填充方法具体的算法流程见"3.2"节.

3.2　实现框架

报文负载字典填充方法的实现框架如图 1 所示,本文将特征串的变型与正常字典中的字符串混合来填充报文负载. 以下是该框架实现的算法流程.

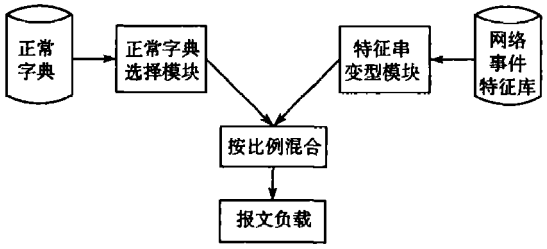

图 1　负载字典构造方法示意图

Fig. 1　Model of constructing payload with dictionary

输入: 字典、待填充报文头部队列、网络事件特征库、特征串变型在负载中所占比例.

输出: 背景流量

实现步骤:

Step1　按照所需生成背景流量的协议,从网络事件特征库中随机选取该协议对应的某一特征串.

Step2　根据"3.1"节给出的变型方法对特征串进行变型,得到特征串变型 A.

Step3　从待填充的报头队列中选择一报头进行填充.

Step4　根据所选报头中给出的负载长度 L,特征串变型在负载中所占比例 R,以及特征串变型 A 的长度 S,计算所需填充的特征串变型个数.

$$N = (L * R)/(100 * S) \tag{2}$$

Step5　从字典中随机选取字符串 B,并根据报头中给出的负载长度裁剪该字符串. 假设报文长度 L,特征串变型的长度 S,特征串变型个数 N,则裁剪后的长度 $T = L - N * S$. 裁剪的方法是从 A 字符串的头部开始截取长度为 T 的子串形成字符串 C.

Step6　将数量为 N 的特征串变型 A 与 Step5 中所得字符串 C 拼接后,形成完整的报文负载. 其中特征串变型的数量由 Step3 中给出.

Step7　将报文的负载与报文头部拼接形成完整的报文.

Step8　如果待填充报头队列为空,程序结束; 否则,返回 Step3.

3.3　实验与评估

为了检验特征串变型填充的效果,作者进行了不同填充比例的比较实验. 实验主机为双 Xeron2.4 GB 处理器,1 GB 内存,本文选用入侵检测系统 SNORT 的字符串匹配模块为实验对象,版本为 2.2.0. SNORT 的字符串匹配模块实现了 B-M 算法. 针对该算法,得出了估计匹配算法进行字符比较次数的公式. 假设比较次数为 C,填充总长度为 L,特征串个数为 N,特征串平均长度为 S,填充比例为 M,则

$$C = (L * (N - 1))/S + (L * M)/100 + ((100 - M) * L)/(100 * S) \tag{3}$$

本文从 SNORT 规则库中提取了 FTP 攻击系列特征串,在其中本文选取特征串"MDTM"作为变型的原型,并根据上文所述的方法变型为"ADTM". 具体的实验流程为

(1) 随机生成 10 万个 500 字节长度的字符串.

(2) 检查生成的字符串与所提取的特征串是否有冲突; 如果有冲突,重新生成该字符串.

(3) 根据设定的比例把特征串变型填充入这些字符串.

(4) 调用 SNORT 的字符串匹配模块检测这些字符串,统计字符串匹配模块进行字符比较的次数. 统计结果如图 2.

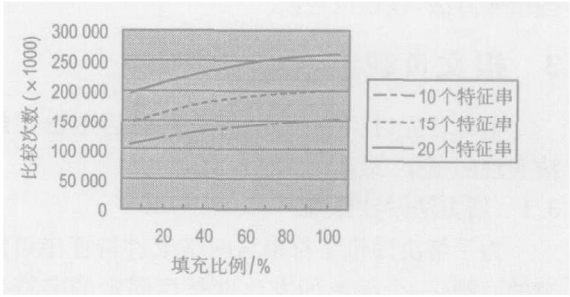

图 2　负载填充比例与字符串匹配模块比较次数变化曲线

Fig. 2　Relationship between proportion of payload and complexity of string pattern matching module

实验表明,随着特征串变型填充比例的增加,SNORT 字符串匹配模块进行字符比较的次数明显增加,并且比较次数与公式基本相符. 由此可见,本文所提供的方法能有效增加背景流量对被测系统的压力,并且压力的大小可控.

但是特征串变型的加入也将影响测试系统的性能. 在填充负载的时候除了正常字符串复制, 还要增加特征串变型的复制. 而在存储报文的时候, 也要增加特征串的索引和位置. 但这些变化对系统的性能影响并不大.

4　结　语

本文设计了背景流量报文负载的字典填充方法, 字典的内容由通过验证不会引入冲突的正常字符串组成. 在需要填充报文的负载时, 从字典中随机选取字符串进行填充, 并通过插入一定比例的网络事件特征串变型来调节报文对被测系统的压力. 这种方法避免了背景流量引入误报. 经实验证明能有效控制报文对被测设备产生的压力, 所生成的背景流量能针对需要考查报文负载语义的网络流量检测设备(如 NIDS 、应用层防火墙等) 进行更合理的测试. 在测试系统的性能方面, 使用字典方法进行报文负载填充的速度比随机串填充方法快了 400 倍, 而所需存储背景流量的空间也有大幅度的减小. 在需要进行压力测试时, 这种构造背景流量方法需要了解被测系统所能检测的网络事件特征和使用的字符串匹配算法, 因此需要被测系统在测试前公开这两项资料.

参考文献:

[1] RANUM M J. Experiences benchmarking intrusion detection systems[EB/OL]. [S. l.]: NFR Security, Inc, 2001[2005-07-05]. http://www.snort.org/docs/Benchmarking-IDS-NFR.pdf.

[2] MELL P, HU V. An overview of Issues in testing intrusion detection systems[EB/OL]. [S. l.]: NIST, 2003 [2005-07-05]. http://csrc.nist.gov/publications/nistir/nistir-7007.pdf.

[3] MCHUGH J. Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory [J]. **ACM Transactions on Information and System Security**, 2000, **3**(4): 262-294.

[4] 汪　洋, 龚　俭. 入侵检测系统评估方法综述[J]. 计算机工程与应用, 2003, **32**: 171-173.

[5] ANTONATOS S, ANAGNOSTAKIS K G, MARKATOS E P. Generating realistic workloads for network intrusion detection systems[A]. Proceedings of the Fourth International Work-shop on Software and Performance [C]. [S.l.]: ACM Press, 2004.

A method of constructing payload of packets in background traffic with dictionary

GONG Jian[1,2], WU Xiong[1,2], YANG Wang[1,2]

(1. Department of Computer Science, Southeast Univ., Nanjing 210096, China;
2. Jiangsu Province Key Laboratory of Computer Networking Technology, Nanjing 210096, China)

Abstract: The background traffic generation becomes an important part for the network test. Nowadays the payload of packets in the background traffic is usually constructed with random strings. Such a method may bring about many problems, such as unexpected test results which can reduce the accuracy of the test, unexpected processing pressure to the tested system, and performance decrease of the testing system when generating the background traffic in a real-time mode. Furthermore, the storage of such random data for offline test requires much of space. To tackle these problems, a novel method is proposed, which uses the variety of the feature patterns whose ware was abstracted from possible traffic to construct a payload dictionary, and the background traffic payload was composed of the dictionary. The experiments show that this method is efficient, and can reduce the problems caused by the random payload.

Key words: background traffic; payload; dictionary; network test