# 基于 AOI 方法的未知蠕虫特征自动发现算法研究*)

顾荣杰[1] 晏蒲柳[1] 邹 涛[2] 杨剑峰[1]

(武汉大学电子信息学院通信工程系 武汉 430072)[1] (北京系统工程研究院 北京 100101)[2]

**摘 要** 近年来频繁爆发的大规模网络蠕虫对 Internet 的整体安全构成了巨大的威胁,新的变种仍在不断出现。由于无法事先得到未知蠕虫的特征,传统的基于特征的入侵检测机制已经失效。目前蠕虫监测的一般做法是在侦测到网络异常后由人工捕获并进行特征的分析,再将特征加入高速检测引擎进行监测。本文提出了一种新的基于面向属性归纳(AOI)方法的未知蠕虫特征自动提取方法。该算法在可疑蠕虫源定位的基础上进行频繁特征的自动提取,能够在爆发的早期检测到蠕虫的特征,进而通过控制台特征关联监测未知蠕虫的发展趋势。实验证明该方法是可行而且有效的。

**关键词** 未知蠕虫,特征自动提取,面向属性归纳

## An Automatic Worm Signature Extraction Algorithm Based on Attribution-Oriented Induction Method

GU Rong-Jie[1] YAN Pu-Liu[1] ZOU Tao[2] YANG Jian-Feng[1]

(School of Electronic Information, Wuhan University, Wuhan 430072)[1] (Beijing Instituet of System Engineering, Beijing 100101)[2]

**Abstract** The frequent explosion of massive worm propagation becomes a huge threaten to Internet security and caused countless losses. The traditional signature based IDS fails to detect new worm due the absence of the ability to detect characteristic of unknown worms. Currently, worm monitoring mainly depends on artificial analysis on the captured worm traffic after the early-bird system detected anomaly worm traffic and put the signature into the high speed detection system. This paper proposed an automatic worm signature extraction algorithm based on Attribution-Oriented Induction method. It can detect worm signature using a Hash method in the early stage of worm propagation and then track the worm spread trend through signature correlation in the control center of system. The subsequent experiment result shows that the algorithm is feasible and effective.

**Keywords** Unknown worm species, Automatic signature extraction, AOI

## 1 背景介绍

基于规则和基于异常的分析是入侵检测系统(Intrusion Detection System)的两大主要检测机制。基于规则的检测机制依赖已知的特征事件集,无法对新的攻击做出检测。而异常检测依赖于对"正常"模式的描述,从而检测出未知的"异常"。蠕虫是一段能够自动运行,并且能够利用远程主机服务脆弱性通过网络进行自身传播的代码[1]。对于入侵检测系统来说,它是典型的未知攻击类型。每一次蠕虫的爆发都给互联网安全带来严重的威胁,并导致了巨额的经济损失。现有的主流检测方法无法有效应对未知蠕虫的进攻和传播,因此对未知蠕虫特征自动提取的研究已经成为当务之急。

## 2 相关工作

经历了几次全球性的蠕虫大爆发之后,蠕虫的研究已经成为网络安全领域的热点问题。传统的基于特征的方法对于大量未知蠕虫的检测已经失效。许多学者提出了新的检测方法。Xuan Chen 等人[2]设计了一个基于路由器监测的蠕虫发现系统 DEWP,通过监测路由器出入口双向流量中被频繁访问的相同的目标端口来发现蠕虫活动,并希望通过进一步流量提取发现其特征,这种方法存在以下问题:(1)需要路由器检测到蠕虫攻击端口相关双向的巨大流量之后才能做出分析,说明蠕虫传播在整体上已经达到泛滥的程度,检测对于控制而言已经失去最佳时机;(2)在蠕虫特征提取中,DEWP 系统分析的是整个骨干网的流量,这对系统无疑会造成巨大的压力。George Bakos[3]等人依赖 ICMP-T3(目标地址不可到达)报文进行蠕虫行为检测,一方面需要在监测网络修改路由器设备配置,使其产生特定的 ICMP-T3 报文,另一方面由于某些网络对 ICMP 消息的抑制导致其方法本身存在不可靠性。此外,UCSD 的 Sumeet Singh 和 Cristian Estan 等人[4]提出了一些蠕虫检测的想法,其中包括借鉴 CAIDA 的 Moore 等人的 Network Telescope 项目思想以及提取部分子串的 HASH 指纹作为未知特征的方式,他们采用的指纹提取算法是 39 位子串的 Rabin 算法。

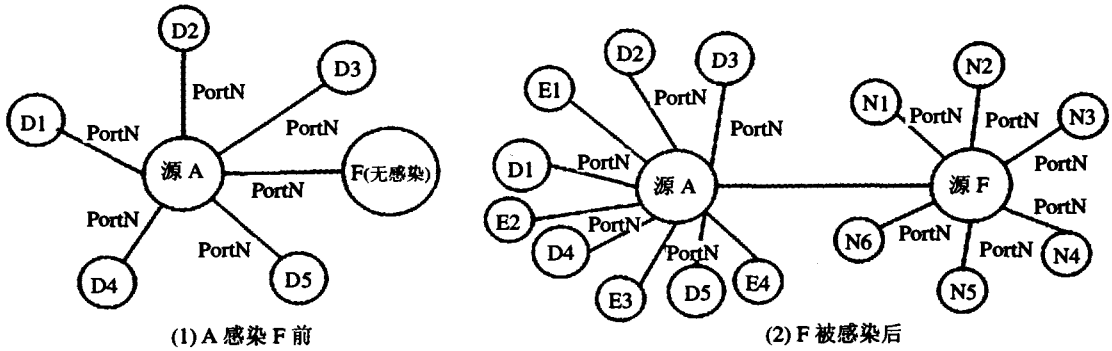
针对 Xuan Chen 等人存在的分析压力的问题,本文采取分而治之的分阶段策略。策略按先后分为三个阶段:先进行快速传播源定位,由此完成针对性的二次流量采集,最后进行未知蠕虫的特征自动发现。本文的前期工作[5,6]通过构建访问兴趣度关系树,使用源地址活跃度、目标地址离散度和响应度准则等多个测度联合约束,以较低的资源代价实现了监测目标网络未知蠕虫的发现,并实现了快速传播源定位,从而缩小了目标搜索的范围,大大降低了分析代价。本文将在此基

*)基金项目:国家自然科学基金项目(90204008)。顾荣杰 博士研究生,主要研究方向包括智能网络信息处理,网络安全等;晏蒲柳 博士,教授,博士生导师,长期从事计算机网络通信,网络管理等方面的研究。

础上进一步提出面向属性归纳(AOI)的频繁集提取算法,实现未知蠕虫特征自动发现。

## 3 蠕虫行为特点与蠕虫源定位

分析蠕虫的典型传播行为,图 1 描述了蠕虫从结点 A 向结点 F 传染前后的模式变化情况。




(1) A 感染 F 前　　(2) F 被感染后

图 1　蠕虫感染前后通信模式特点的变化(结点 F)

图 1(1)中可以看到,源 A 向 D1～D5 及 F 大量发送指向同一端口的攻击数据包,源 A 的通信模式呈现发散状,且其频繁连接的目标地址是相同的,这样的模式称为花束状(Bloom),而 F 此时尚未被感染,其对外通信无花束状模式存在。图 1(2)是结点 F 被感染以后,其通信行为继承了传染源 A 的特点,也呈现出了明显的花束状。蠕虫的传播具有贪婪的天性,一旦成功占领某台主机以后,它会以最快的速度向尽可能多的目标进行扫描扩散。由于同一种类型的蠕虫其所有行为基于同一套攻击、传播机制,因此表现在对外发送的负载内容上呈现大量频繁重复的特点,通过分析频繁重复的负载内容使实现特征自动提取成为可能。综上所述,所有感染了同一类型蠕虫的主机对外传播方式具有自相似性的特点,即:

- 在短时间内产生大量对外通信连接,通信行为具有"花束状"(Bloom);
- "花束状"通信连接的具有水平扫描的特性;
- "花束状"通信的目标地址具有离散分布性;
- "花束状"通信的负载内容具有一致或相似性。

本文的前期工作实现了监测目标网络中的蠕虫传播源快速定位,定位结果如图 2 所示。

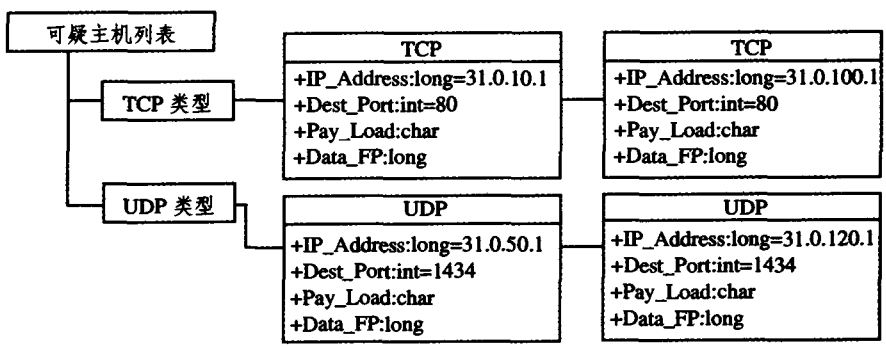



图 2　源定位的可疑蠕虫主机信息列表

从上表可见,定位结果给出了源 IP 地址,蠕虫活动影响的协议名称和端口,在此基础上,我们可以进行二次流量获取,即进一步捕获一段定时区间(默认取 1 分钟)内满足定位结果〈源 IP 地址,协议类型,目标端口〉的流量,目标格式为〈源 IP,目标 IP,端口,负载长度,负载前 64 位的 MD5 指纹〉,作为我们下面分析的基础。

## 4 基于 AOI 方法的蠕虫特征提取算法

### 4.1 蠕虫特征提取问题的特点

蠕虫影响的端口、扫描目标地址与被感染的主机地址之间找不到明显的属性依赖关系,蠕虫对外访问使用的端口和目标地址也可能被正常访问所使用,因此用传统的知识发现方法如粗集理论、关联规则等方法不适于进行特征的发现。其次,从效率上考虑,传统的规则提取算法计算量太大,比如关联规则的候选集生成效率很低,对于骨干网上每秒几万甚至几十万个数据包的速度要求而言显然无法满足要求。因此引入下面的面向属性的归纳方法。

### 4.2 传统的 AOI 方法以及存在的问题

面向属性的归纳方法(Attribute-oriented Induction, AOI)最早于 1989 年首次提出,Cai, Han 及 Nishio, Kawano 等学者相继对其进行了进一步的研究和扩充[7~9]。面向属性归纳方法的基本思想是[10]:通过考察任务相关数据集中每个属性的不同值的个数进行数据概化,聚集通过合并相等的广义元组,并累计它们对应的计数值。AOI 的计算结果最终以频繁规则提供给用户。AOI 方法与常规的规则发现方法相比,由于没有复杂的计算,不需要产生中间候选集,直接对属性进行概化与归纳,在处理速度上已经有了很大的提高,但是仍然存在以下问题:

• 传统的 AOI 算法中，在所有属性$\{A_i, i=1,2,\cdots,N\}$中概化属性 $A_i$ 的选择标准和策略没有限制，意味着需要经过尝试才能知道是否需要概化，同时属性概化阈值 $r$ 仅关心不同属性值的个数，而对某个属性取值本身对原始记录集的覆盖度并不关注，有时候会带来过度概化（over generalization）[11]的问题。

• 传统的 AOI 算法没有对归纳过程的结束条件进行约束，算法对数据进行过度规则挖掘，出现大量有效但无用的规则，也大大耗费了资源和时间。

### 4.3 基于频繁度约束的改进 AOI 方法

根据上面的分析，同时结合我们的应用需要，我们对 AOI 算法的相关策略进行修改，改进过程遵循三个原则：1）规则必须是“频繁”的，频繁规则的定义是：如果满足规则 $P_i$ 原始记录数大于设定的阈值，则称规则 $P_i$ 为频繁规则；2）规则在“频繁”的同时，保持更多的细节信息，即尽量不被概化到更高层次；3）合理控制归纳的程度，在发现主要规则后及时停止。

令 AOI 算法的输入任务数据集为 $W_0=\{A_1,A_2,\cdots,A_N,C\}$，$\{A_i, i=1,2,\cdots,N\}$是数据表属性，$C$ 为记录的计数，初始化时，该字段为 1。记 $a\in W_0$ 为任务数据集的记录。如果两条记录 $a,a'$满足 $a.A_i=a'.A_i(i=1,\cdots,N)$，则称$a,a'$相等，其计数属性可以不同。

基于频繁度约束的改进 AOI 算法丢弃了传统 AOI 算法中以属性概化阈值 $r$ 和概化关系阈值 $d$ 为约束的概化条件判断，改用频繁度评价的方法，并增加了归纳进度控制，从而避免了过度概化问题和大量无用规则的产生，算法流程如下：

• 第一步：记 $T=W_0$，首先扫描数据库，对任意两条相等的记录 $a,a'$进行合并，计数值 $C=C_a+C_{a'}$；

• 第二步：寻找所有 $a_i\in W_0$，如果$\exists a_i$ 满足 $a_i.C>\min\_size$，则直接将其输出为规则，其支持度为 $Support(a_i)=a_i.C/T.countall$，记剩下的数据集 $T'=T-a_i$，重复本次操作，直到不再有 $a_i.C>\min\_size$ 为止，接下来才进行概化，这在一定程度上避免了对频繁规则的过度概化问题；

• 第三步：概化变量选择及概化过程。为了选择合适的属性 $A_i$ 进行层次概化，在余下的数据集 $T'$中，对每一个属性 $A_i$，计算其最频繁的一个取值 $v$，满足 $F_i:=\max\{f_i(v)\mid v\in D_{A_i}\}$，其中，$f$ 为 $v$ 的计数累加值，进一步令 $F=\min(F_i)$，$i=1,\cdots,N$，则相应的 $A_i$ 就被选择进行概化。我们的规则归纳是按频繁度递减进行的渐近式搜索过程。我们从最频繁的一系列属性取值中找出取值最少的一个，对其进行概化，并以此为标准对其它的频繁属性进行属性归纳，这样处理考虑到两个因素：1）如果某规则能够被归纳为频繁规则，该规则所包括的任一属性的最小值必然大于 $\min\_size$。因此我们的归纳必须针对最离散的属性进行，这也是为了避免过度概化问题的发生；2）通过对 $F_i$ 最小的属性进行概化，将其提升到一个比较高的层次，从而使其不再成为 $F_i$ 值最小的属性，使整体向频繁规则方向靠拢。对 $F_i$ 最小的属性进行概化后，重新计算，$F=\min(F_i)$，$i=1,\cdots,N$，进而按第三步的原则选择新的属性，再次对数据集进行概化操作。如果某个属性 $A_i$ 遍历概化树之后仍不能被有效概化（属于不能被概化类型），则在后续归纳过程中删除该属性。

• 第四步：算法进行归纳进度控制。每次概化后对所有属性计数进行重新扫描，如存在新的规则 $a_n\in T'$满足 $a_n.C>\min\_size$，则将 $a_n$ 输出为新的频繁规则，并令 $T'=T'-a_n$，继续进行第三步操作。当第 $j$ 次输出规则 $a_j$ 时，计算累计支持度

$$s=\sum_{i=1}^{j}Support(a_i)=\frac{\sum_{i=1}^{j}a_i.C}{T.count} \qquad (式\ 1)$$

当 $s>\min\_sup$ 时（$\min\_sup$ 是用户定义的算法终止时的累计支持度阈值），停止 AOI 归纳，报告发现的规则并退出算法循环。

### 4.4 基于改进 AOI 算法的蠕虫传播特征析取算法

对于蠕虫传播特征的提取，首先生成需要分析的工作数据集 $W$，接下来分为五个步骤：

• 第 1 步，属性删简。

与 DoS 模式不同，蠕虫跟踪针对的是源地址和目标端口，因此基于源地址和目标端口两个属性划分的信息增益为 0，根据信息增益原则，先将该二属性从运算过程中排除。

• 第 2 步，设置频繁度阈值 $\min\_size$ 和累计支持度阈值 $s$。

在 Worm 特征提取算法中，频繁度阈值 $\min\_size$ 被设定为随数据集大小 $C_{size}$ 动态调整，$\min\_size=C_{size}\times 10\%$，这有利于提高算法的适应性。定义 TCP 累计支持度阈值 $s$ 设定为 50%，UDP 类型为 85%，两者的差别是因为 TCP 随机扫描中存在响应率问题影响频繁规则支持度，而 UDP 传播不需要建立会话，因此可以监测到大量特征数据包。

• 第 3 步，设置概念结构树和概化操作映射。

在蠕虫的对外扫描中，存在某些概率上的特征，通过对目标地址的概化分析可以了解蠕虫传播的策略特点。红色代码 II 的扫描策略为：随机产生掩码为 0.0.0.0 的地址，大概占 12.5%，产生和本机地址为同一个 C 地址掩码为 255.0.0.0 的地址，大概占 50%，剩下的产生同一个 B 的地址。蠕虫目标地址概念结构树如图 3 所示。定义的映射关系为：$f_1=f_2\mid ip'\leftarrow ip\gg 8$。图中概化 IP 地址后面的/8 和/16 参数代表子网掩码。

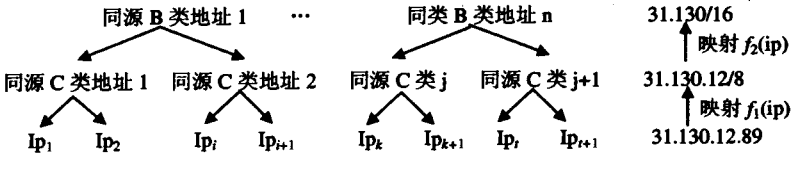



图 3 IP 地址的概念结构树

当概化过程已经遍历完概念树（即二次概化后）仍未产生频集时，说明目标地址的选择是非聚集分布的，算法将自动删除该目标地址属性。

## 5 数据分析

本文的数据取自某拥有超过 3,500 台主机的政务网。原

始数据通过其核心交换机 Cisco Catalyst 6509 的 Spanning 方式采集得到，本文设计的数据包捕获核心模块 Flowcapture 采用 winpcap 库，在完成 TIR 树建模的同时，输出原始流到日志数据库。

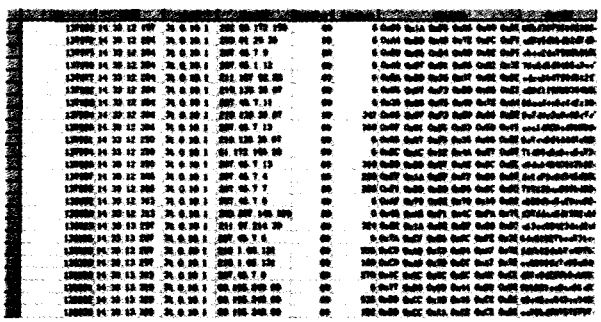
经前期定位检测，得到如下可疑地址信息列表。

表 1　定位检测结果

| 可疑地址 | 协议 | 目标端口 | 包长 |
|---|---|---|---|
| 31.0.100.1 | TCP | 80 | N/A |
| 31.0.10.1 | TCP | 80 | N/A |
| 31.0.50.1 | UDP | 1434 | 300～400 |
| 31.0.120.1 | UDP | 1434 | 300～400 |

对 31.0.10.1 和 31.0.100.1 的 TCP80 端口流量进行 1 分钟的二次获取，得到如下数据集。

表 2　二次流量获取结果集



31.0.10.1 是一台很典型的 HTTP 代理服务器，因此正常情况下对外的访问请求数量比较大，平均达 9,800Pkts/分钟左右，加上蠕虫活动数据，总的对外请求数为 17,569 次。可见，蠕虫的活动并没有超过其正常应用的活动数量。对我们的算法而言如果能从不占主体的活动记录中找到活动的蠕虫特征，则表明算法能适应复杂的应用环境。因此用它来检测算法的有效性是比较典型性的。算法首先扫描各个数据字段(DestIP，datalen，tcpflag，pktdata，dataFP)，找出各个字段最频繁的项及其对应的频繁度。对于本例，DestIP 字段的最大频繁项集的频繁度最低为 36，datalen 的最大频繁项为 0，对频繁度值为 7275，因此，根据启发性搜索原则，首先对其进行一级概化，即进行 $f_1|ip'\leftarrow ip>>8$ 概化映射而变为同源 C 类地址。概化映射结束后再次进行 DestIP 字段的频繁度计算，得到最大频繁项集的频繁度为 125，可见其频繁度与概化前相比有了提高，此外，记录与源 IP 同源的 C 类地址 31.0.10/24 的出现频度为 113 次。虽然概化后 DestIP 的最大项集频繁度上升，但在所有字段中仍是最小的。因此，继续进行第二次概化，即通过 $f_2|ip'\leftarrow ip>>8$ 概化映射变为同源 B 类地址，进一步计算其频繁度，得到最大频繁项集为 31.0/16，对应频繁度为 3041，其所占比例为 3041/17569≈17.31%。从抽样来看对同源 B 类地址的扫描比例是比较突出的，这与红色代码蠕虫的扫描特性是相符的，理论比例应该为 37.5%，由于该主机本机的活动量比较大，蠕虫的活动比例数值相对偏小了。两次概化以后，DestIP 已经到达概念树顶，不能继续被概化。接下来，统计相应的频繁度，DestIP 因为最终频度无法达到总数据集大小的 20%而被删除。随后在(datalen，tcpflag，pktdata，dataFP)四个属性中进行频繁项选取，即对所有记录进行属性值扫描，合并所有属性一致的记录。经归纳最终生成两条规则如表 3，4 频繁度依次为 7275 和 3571，其累计频度 61.73%大于 50%，算法停止。第一规则的支持度为 7275/17569≈41.4%，第二规则的支持度为 3571/17569≈20.33%＞最小支持度阈值 20%，因此是有效的。如果找到的第一规则的数据长度为 0，系统将自动取第二大规则作为蠕虫的特征。如果分析中不能发现有一条规则其支持度＞最小支持度阈值 10%，则说明该数据集中无频繁规则存在，算法返回。对 31.0.10.1 和 31.0.100.1 的分析汇总在下表中，由于 31.0.100.1 不是 HTTP 代理服务器和网关，并且对外的 HTTP 业务活动较少，因此蠕虫的活动相对比较明显，其特征的支持度较 31.0.10.1 有了一定幅度的提高。

表 3 是对 31.0.10.1 进行 AOI 提取特征规则的结果。

表 3

| ID | 规则描述(srcIP，destPort，dataFP) | 支持度 | 说明 |
|---|---|---|---|
| 1 | 31.0.10.1，TCP，80，NULL，NULL | 41.4% | 负载为空，不报告 |
| 2 | 31.0.10.1，TCP，80，cfb 12029bd 565e 2c 442ed 6df 839848d0 | 20.33% | 红色代码 II 蠕虫特征 |
| 3 | 31.0.100.1，TCP，80，NULL | 72.53% | 负载为空，不报告 |
| 4 | 31.0.100.1，TCP80，cfb1 2029bd 565e 2c 442ed 6df 839848d0 | 22.4% | 红色代码 II 蠕虫特征 |

对主机 31.0.50.1 和 31.0.120.1 进行同样分析，得到如下结果。

表 4

| ID | 规则描述(srcIP，destPort，dataFP) | 支持度 | 说明 |
|---|---|---|---|
| 1 | 31.0.50.1，UDP，1434，abead 3d0953acac 112cb 116873108454 | 94.24% | Slammer 特征 |
| 2 | 31.0.120.1，UDP，1434，abead3d 0953acac 112cb 116873108454 | 97.18% | Slammer 特征 |

最后进行检验，指纹“cfb 12029bd 565e 2c 442ed 6df 839848 d0”对应的 16 进制串为“0x2F 0x64 0x65 0x66 0x61 0x75 0x6C 0x74 0x2E 0x69 0x64 0x61 0x3F 0x58 0x58 0x58…0x58”，换成 ASCII 字符串为“/default.ida? XXXXXXXXX…XXXX”，正是红色代码 II 蠕虫的特征请求字符串[12]。而指纹“abead3d0953acac112cb116873108454”对应的 16 进制串为“0x04 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01…0xAE 0x42 0x90 0x90 0x90 0x90 0x90 0x90 0x90 0x90 0x68 0xDC 0xC9”，也正是 SQL Slammer 蠕虫的特征请求字符串[13]。由于 UDP 攻击不需要 SYN 握手连接，也不需要响应等待，因此攻击的时间密集度大大提高。另一方面，Slammer 蠕虫的作者在设计时精简了代码，将 376 字节的汇编攻击代码封装在单个 UDP 数据包中发起攻击扫描，其内部循环非常简短，这些条件使它成为有史以来传播最快、对互联网攻击影响最大的蠕虫[14]。从实验结果可以看到，算法对 Slammer 蠕虫的特征归纳提取非常有效，其特征的支持度达到了 94%以上。验证结果表明，基于本文的算法有效地检测到了传播的蠕虫特征。

(下转第 137 页)

$K_{ch}$(GM向每个用户分发的一种组密钥)而秘密使用的加密密钥。在注册阶段得到的$(\overline{x_j},\hat{x}_j)$是成员$u_j$秘密使用的解密密钥。

### 5.2 局限性分析

$A$的计算可简化:由指数律及YVZ算法中$g_i$的定义有$A=g^{A(n)}$,其中$A(n)=\sum_{j=1}^{n}\sum_{i=0}^{n-1}a_i x_{ij}$。但因$\sum_{i=0}^{n}a_i x_{ij}=f(x_j)=0$及$a_n=1$,故$A(n)=-\sum_{j=1}^{n}x_j^n$。又因$g\in Z_p^*$,故我们可用Fermat小定理来避免求模逆元$g^{-1}(\mathrm{mod}\ p)$:按$e(n)\leftarrow(p-1)-(\sum_{j=1}^{n}x_j^n)(\mathrm{mod}\ p-1)$及$A\leftarrow g^{e(n)}$来计算A(还可利用著名的平方-乘算法[6]等技巧来加速计算)比原算法开销更低。因为原算法需要$2n$次求幂运算和$2(n-1)$乘法运算,而上述算法仅需$n+1$次求幂运算、减法2次及1次模运算。

CV方案的局限性主要体现在同余方程$x^2\equiv 1(\mathrm{mod}\ q)$的解数$|S|$($S$含元素的个数)不能满足组播应用的实际需求。事实上,因$x\equiv\pm 1(\mathrm{mod}\ q)$为两个平凡解,故$|S|\geqslant 2$。但因$Z_q$是有限环,其二阶元的个数$|S|$必有限。文[5]从整数集$Z$的角度断言$|S|=\infty$(从而选出$n$个解以构造在加密过程$c\equiv K_{ch}A^{*}(\mathrm{mod}\ p)$中所需要的$s$,其中$k\in Z_q$是GM选取的随机数)。这既与同余方程的解数概念相违,也忽视了$s$在加密过程中由Fermat小定理所提供的角色,即$s$与$s\ \mathrm{mod}(p-1)$的作用是等效的。

特别地,当$q=2^l$时,有$|S|=4$[10],即此时系统能容纳的最大成员数$n=4$;这远远不能满足大型组播通信的应用需求。同时,小的$|S|$缩小了敌手穷举攻击$s$的范围,从而埋下安全隐患。

若允许$s_i$重复取值,则这又造成CV方案中的两个信道密钥更新方案均丧失前向/后向保密性。

**结论** YVZ算法及其改进G算法的基本安全性都是基于DLP的难解性,但后者不但省去了不必要的强单向散列函数,保持了理想的语意特性,而且通过恰当选择$p-1$的素因子还能以高概率抵抗零因子攻击。

基于YVZ算法的CV方案的局限性在于同余方程的解数不能安全地满足大型组播应用的实际需求。利用G算法构造安全而实用的组密钥管理方案是我们下一步的研究工作。

## 参考文献


1 Rafaeli S, Hutchison D. A Survey of Key Management for Secure Group Communication [J]. ACM Computing Survey, 2003, 35 (3): 309~329

2 Dondeti L R, Mukherjee S, Samal A. Scalable Secure One-to-Many Group Communication Using Dual Encryption [J]. Computer Communications, 2000, 23(7): 1681~1701

3 Hardjono T, Dondeti L R. Multicast and Group Security [M]. Norwood, MA: Artech House, INC, 2003

4 Yi M, Varadharajan V, Zhao W. A Robust and Secure Broadcasting Scheme [C]. Proceedings of IndoCrypt'2001, Lecture Notes in Computer Science, Springer Verlag LNCS Series, 2001

5 Chaddoud G, Varadharajan V. Efficient secure group management for SSM [C]. In: 2004 IEEE International Conference on Communications(Paris, France, 20-24 June 2004), Piscataway, NJ, USA, 2004. 1436~1440

6 Stinson D R 著. 密码学原理与实践(第二版). 冯登国 译[M]. 北京: 电子工业出版社, 2003

7 Menezes A J, van Oorschot P, Vanstone S. Handbook of Applied Cryptography [M]. Boca Raton: CRC Press, 1997

8 Lenstra A K. Computational Methods in Public Key Cryptography [M]. In: Niedrreiter H. ed. Coding Theory and Cryptology, Singapore University Press and World Scientific Publishing Co Pte Ltd, 2002

9 Mao W 著. 现代密码学理论与实践[M]. 王继林, 等译. 北京: 电子工业出版社, 2004

10 柯召, 孙琦. 数论讲义(上册). 第二版[M]. 北京: 高等教育出版社, 2001

11 潘承洞, 潘承彪. 初等数论[M]. 北京: 北京大学出版社, 1992


(上接第130页)

根据前面的分析,最后生成的蠕虫特征报告一般具有如下形式:〈srcIP, ProtocolType, destPort, DataFP〉。在控制中心,将分布在骨干网的各个出入口引擎上报的数据在一定时间间隔内将生成一个告警列表,通过对〈目标端口,负载指纹〉关联信息的快速归纳,就可以了解本地监测网中,有哪些地址已经感染了某种特征的蠕虫,这样就可实现基于感染主机数量的统计,有助于中心了解和掌握某种未知蠕虫的传播和发展情况。

**结束语** 蠕虫的检测和防御是一项长期的工作。现有的特征检测手段永远落后于蠕虫的发展速度,因此检测系统需要具备对未知蠕虫特征的自动发现能力。本文基于相关的工作基础,提出了基于改进频繁度约束的AOI算法的蠕虫特征自动提取算法,初步的实验结果表明,基于本文的算法能在蠕虫传播的早期有效地检测到未知的蠕虫特征。在今后的工作中我们将进一步对其在高速骨干网上的应用性能进行测试并加以改进。

## 参考文献


1 Spafford E H. The Internet Worm: Crisis and Aftermath. Communications of the ACM, 1989, 32(6): 678~687

2 Chen Xuan, Heidemann John. Detecting Early Worm Propagation through Packet Matching: [Technical Report ISI-TR-2004-585]. USC/Information Sciences Institute, Feb. 2004

3 Bakos G, Berk V. Early Detection of Internet Worm Activity by Metering ICMP Destination Unreachable Messages. In: Proceedings of the SPIE Aerosense, 2002

4 Singh S, Estan C, Varghese G, Savage S. The EarlyBird System for Real-time Detection of Unknown Worms: [Technical Report CS2003-0761]. UCSD, 2003

5 Gu RongJie, Xia DeLin, Yan PuLiu. An Adaptive Internet Backbone Malicious Activies Detection System Based on Frequent Pattern Mining. [J] GESTS International Transactions on Computer Science and Engineering, 2005, 12: 141~148

6 顾荣杰. 宏观网络安全监测分析方法研究: [博士论文]. 武汉大学, 2005

7 Cai Y, Cercone N, Han J. Attribute-Oriented Induction in relational databases. In: G. Priatetsky-Shapiro and W. J. Frawley, eds. Knowledge Discovery in Databases, Cambrige, MA: AAAI/MIT, 213~228

8 Han J, Cai Y, Cercone N. Data-driven discovery of quantitative rules in relational databases. IEEE Trans. Knowledge and Data Engineering, 1993: 29~40

9 Han J, Fu Y. Discovery of multiple-level association rules from large databases. In: Proc. 1995 Int. Conf. Very Large Data Bases (VLDB'95), 1995

10 Han Jiawei, Kamber Micheline. Data Mining Concepts and Techniques. [M] Canada: Morgan Kaufmann Publishers

11 Julish K, Dacier M. Mining Intrusion Detection Alarms for Actionable Knowledge. In: The 8th ACM Int Conf. on Knowledge Discovery and Data Mining, Edmonton, 2002

12 Shannon M D, et al. Code-Red: Acase study on the spread and victims of an Internet worm. In IMW, 2002

13 SQL Slammer Binary Code. http://www. xfocus. net/tools/200302/384. html

14 Moore D, Paxson V, et al. The Spread of the Sapphire/Slammer Worm, CAIDA, ICSI, Silicon Defense, UC Berkeley EECS and UC San Diego CSE, 2003