

利用分治策略实现 DDoS 攻击路径标识题

翟继强, 唐远新, 叶 飞, 谢怡宁

(哈尔滨理工大学 计算机科学与技术学院, 黑龙江 哈尔滨 150080)

摘 要: 针对分布式拒绝服务(DDoS)攻击防御中的数据包标记溯源技术, 基于分治策略提出用简单的递归关系分别表示攻击树构、攻击路径频率检测、数据包和路径关联这 3 个基本防御实现, 使用单数据包带内路径标识符用于唯一的数据包和路径关联, 而把攻击树构建及攻击路径频率作为独立的带外操作来处理. 通过在真实网络拓扑环境下对网络流量和存储开销的测试, 证明本方法可以高流量负荷下以很小的开销同时对大量受害主机的提供单一数据包回溯保证.

关键词: DDoS; 分治; IP 溯源; 包标记; 攻击树

中图分类号: TP393 **文献标志码:** A **文章编号:** 1007-2683(2014)04-0076-07

Implementation of DDoS Attack Path Labeling Using Divide-and-conquer Strategy

ZHAI Ji-qiang, TANG Yuan-xin, YE Fei, XIE Yi-ning

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: Focusing on packet marking traceback technology in DDoS attack defense, a novel approach is proposed to represent three elementary defending implementations namely attack tree construction, attack path frequency detection and packet to path association as succinct recurrence relations based on divide-and-conquer strategy, which uses single packet in-band path identifiers for unique packet to path association while handles attack tree construction and attack path frequency detection as independent out-of-band processes. By testing and evaluating the network traffic and storage overhead on real-life Internet, the results show that this method realistically achieves single packet traceback guarantees with minimal overhead for a large number of victims under heavy traffic loads simultaneously.

Key words: DDoS; divide-and-conquer; IP traceback; packet marking; attack tree

0 引 言

因为 DDoS 攻击的隐蔽性、分布性和高破坏性使它成为现今 Internet 网络所面临的重要安全威胁之一^[1], 而追踪攻击包经过的路线、构造攻击路径、

直至追溯到攻击者, 可为预防和纾缓 DDoS 攻击提供依据, 从而更有效抗应对 DDoS 攻击^[2].

基于数据包标记溯源(packet marking)^[3]的 DDoS 攻击防御操作可以定义为 3 种:

1) 攻击树构建: 即构造路由器层网络攻击树抽象图的过程, 攻击树的根节点是受害主机或网络, 不

收稿日期: 2013-12-18

基金项目: 黑龙江省教育厅科学技术研究面上项目(12531121).

作者简介: 翟继强(1972—), 男, 教授, 博士, E-mail: zaijiqiang@126.com;

唐远新(1972—), 男, 副教授;

叶 飞(1972—), 男, 副教授.

同的攻击源或对应的入口路由器为叶子节点。

2) 攻击路径频率检测: 是获取加权攻击树的过程, 攻击树的每个边的权重用指定时间内通过该路径的数据包数目来标注, 攻击路径频率可以有效区分合法或非法流量源, 为随后数据过滤的提供可靠依据。

3) 数据包与路径关联: 定义为将数据包与攻击树中某个特定的数据流量源也就是路由路径进行关联的过程。

现有的攻击树构建方法是需要中间路由器确定性(deterministic) 或者概率性(probabilistic) 发送带内(out-of-band) 带外(in-band) 路径标记信息, 然后利用路径构造算法, 对不同的数据包标记(本文称为路径标记分片) 进行提取、整理和校对构建路径攻击树。由于每个路由器要标记多个数据包, 多个路由器之间就可能存在命名空间冲突的问题, 使路径标记分片不可靠, 从而导致低效甚至无法构造攻击树^[4]。而攻击路径频率检测一般是通过统计来自不同中间路由器上路径标记分片数目来间接推算, 但由于数据采集的概率性和相应构建出的攻击树的不可靠性, 经常会导致攻击溯源产生高误报和漏报^[5]。同时横跨多个数据包的指纹分片需要受害主机拥有多个缓冲区来聚合路径信息, 由于不同数据包标记分片间缺乏有效的关联, 可能导致路由器指纹失序, 无法将数据包和攻击路径有效关联, 也无法正确的识别出攻击路径^[6]。

传统的包标记算法存在不确定性、重复标记、收敛性差等缺点, 同时基于包标记的3个基本操作缺乏明确的职能划分, 一个特征指标要同时满足多种需求, 降低了整体性能。因此本文提出使用分布式分而治之(divide-and-conquer) 策略^[7]用简单的递归关系作为基本模块分别表示上述三种不同的实现, 对网络图中的单一路径标记分片直接进行识别。使用单数据包带内路径标识符用于唯一的数据包和路径关联, 而把攻击树构建及攻击路径频率作为独立的带外操作来处理。将上述问题分成了完全独立的问题进行解决, 用简单的递归关系实现各自操作, 这样一来不仅降低了问题的复杂性, 同时也提高了处理效率。

1 攻击树构建

定义一个四元组 $T(n, l, d, p)$ 来表示一颗攻击树, 其中: n 代表路由器的总数; 即树的节点数; l

代表平均跳数, 即树的深度; d 代表路由器的度数, 即树节点的度; p 代表路由路径数, 也就是叶子数。式(1)和式(2)反应了模型树与平衡 K 叉树之间的映射关系。

$$p = d^l \quad (1)$$

$$n = \frac{-d(d^l - 1)}{d - 1} = \frac{\sqrt[l]{p(p-1)}}{\sqrt[l]{p} - 1} \quad (2)$$

为便于分析, 依据文[8]设定假定 d 和 l 的网络测量值分别为 $d \approx 4, l \approx 16$ 。在今天的 Internet 中, 网络终端主机数远远超出了路由器的使用数目, 将全球独一的 IP 地址(或其散列分段)用作路由器标识符导致命名空间极度发散, 见式(3)所示。

$$F_{\text{global}} = l \lceil \log_2(n) \rceil \approx 512 \text{ bits} \quad (3)$$

本文提出只将路径标识符分配给流量源(叶节点), 将问题控制在一定范围内, 从而提高压缩比, 见等式(4)。

$$F_{\text{proposed}} = \lceil \log_2(p) \rceil \approx 32 \text{ bits} \quad (4)$$

攻击树是以受害主机为跟节点的基于不同数据包流的路由器节点级网络抽象图。该图在相对短的时间内会保持静态。为了使受害机能够对不同的攻击源和攻击路径有一个全局性的掌控, 攻击树上的路径插入需要快速完成, 而路径的删除可以相对滞后。攻击树很少或以中断驱动的模式被刷新, 并且只在发生关键性结构变化时, 才会刷新攻击树。通过重载带内数据包标记来构建攻击树可能会导致单个数据包的重复性传输, 因此采用带外数据包标识。

1.1 递归方法

使用分治递归处理方法, 将受害机攻击树的构建问题分为多个子问题, 分散到每一个路由器上, 分别由路由器的邻居(攻击树的孩子节点)进行处理。然后将每一个子问题的解决整合在一起, 从不同的流量源将攻击树向上递归至受害机。采用的是自底向上的方法, 而非传统由受害机控制的自顶向下的方法。每一个路由器聚合它的直接邻居的攻击子树(T_{R_i}), 转发给它的直接上游邻居(父节点)。这样就以自低而上的分布模式逐步完成了攻击树的演变。

1) 逻辑表示

如下图1所示的攻击树中, 路由器 R_1 有4棵不同的子树 R_2, R_3, R_4 和 R_5 。

式(5)给出了攻击子树的逻辑表示, 其中 D_{R_i} 和 T_{R_i} 分别代表路由器 R_i 的度数和子攻击树。式(6)是攻击树中每个路由器的通用表达式, Δ_{R_i} 代表 R_i 的直接子树。

$$T_{R_1} = D_{R_1} \cup T_{R_2} \cup T_{R_3} \cup T_{R_4} \cup T_{R_5} \quad (5)$$

$$T_{R_i} = D_{R_i} \cup_{R_j \in \Delta R_i} T_{R_j} \quad (6)$$

利用简介的递归关系实现了提出的分布式分而治之方法,这样的攻击树构建能够确保每个路由器不产生冗余信息,所需运算量也小,可以获得整个攻击子树的全局视图.

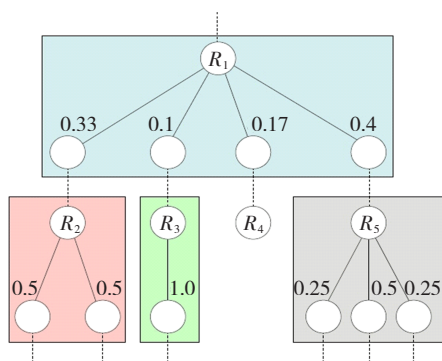


图1 模块化攻击路径树

2) 物理表示

路由器 R_i 的攻击子树的物理表示如式(7)所示,其中 λ 表示因抽象而未能显现的进一步递归扩展.

$$T_{R_1} = 4 \cup 2\lambda\lambda \cup 1\lambda \cup 0 \cup 3\lambda\lambda\lambda \quad (7)$$

如果使用节点的度来标记攻击树的每个节点,那么式(7)还表示攻击树的前序遍历. 利用前缀波兰表示法中重建原始 k 叉树的技术^[9],在已知所有中间节点的元组的情况下,使用路由器的度来标识每个路由器的攻击子树,受害主机或中间路由器可以相对容易地从攻击子树的前序遍历重构攻击树.

这种递归关系的优势体现模块化,如图1的不同阴影区所示,攻击树任何的结构变化都支持简单的即插即用设计,从树叶扩展至树根,不需要对重构整个攻击树,也不应影响其他独立子攻击树. 通过周期更新或只有当攻击子树发生结构性变化时来触发更新,可以很好的模拟因特网的动态路径特征.

1.2 攻击树修剪

式(8)表示根自受害主机攻击树的最大尺寸,对于谷歌与雅虎之类的高流量门户网站而言,攻击树的尺寸可能达到无法控制的状态. 必须使用树修剪技术(XR_i) 动态消减攻击树的尺寸到可控的水平.

$$\text{TreeSize}_{\max} = n \lceil \log_2(d) \rceil \quad (8)$$

由于攻击树的尺寸规模随着 n 或者 p 线性增长,我们选择限制攻击树中主动跟踪流量源的数量. 式(9)表示在网络中有多个中间路由器支持修剪的

新的递归关系.

$$T_{R_i} = X_{R_i}(D_{R_i} \cup_{R_j \in \Delta R_i} T_{R_j}) \quad (9)$$

这种修剪机制的分布性特征使得不同的 ISP 能够使用独立的策略,确保树修剪的本地最优性. 实际应用中树修剪还可以融入其他的设计参数,如相对资源利用率、受害主机的反馈、分布式监控系统的反馈、随机选择路径的轮询调度,使用本地特征库的黑或白名单等.

攻击树修剪是可选项,不会提出的攻击树结构技术造成不利影响. 合理使用数修剪可以带来很多好处,如在攻击树模型 $T(n, l, d, p)$ 中,设定 $p \approx 1\,000\,000$,而 $l \approx 16 \Rightarrow d \approx 2.4$,可以推导出同时监测 $1\,000\,000$ 个攻击流量源的最大攻击树尺寸只有 507 KB.

2 攻击路径频率检测

在传统的 DoS 攻击期间,一般可以很容易发现和识别持续超高流量的单一攻击源. 但 DDoS 一般利用大规模僵尸网络进行协同攻击,每个攻击源只可以使用中低流量,并且面向特定目标的网络热点活动有时也会产生一个超出容限的突发流量尖峰造成主机和网络出现拒绝服务现象,这样就很难单纯基于流量来分合法和非法数据源^[10]. 此时 DDoS 攻击防御问题就变成对超出一定阈值的不同路径的流量进行排序的问题,排序划分的粒度将由高负载流量下的目标机器和网络来确定. 流量排序需要数据包的指标刻度,如负载特征、协议和标志位等,也需要路径层的数据指标刻度,如频率、流脉冲特征、源掩码等. 本文主要专注于路径层指标,因为数据层指标可以很容易从数据包中收集.

上节中使用带外包标记而得到的攻击树实际上就是个攻击路径树,仅嵌入了路由器的连接信息. 我们将路径频率分布编码嵌入到原始攻击树,采取哈夫曼编码和平衡圆括号嵌入技术^[11],构建一个新的攻击路径频率树.

2.1 频率编码和嵌入

哈夫曼编码是一种熵编码数据压缩算法,是将可变长度的前缀代码分配给符号,从而使代码长度和符号概率相匹配. 对于经过排序的 m 个符号概率集合,用一个节点替换两个频率最低的符号节点,新节点的频率就是这两个节点的频率之和,这样新节点就是两个原符号节点的父节点了,依次循环,线性时间哈夫曼代码生成器就使用自下而上的树构造机

制将不同的符号映射成高效二进制代码而组成 $2m-1$ 节点的二叉树. 哈夫曼编码提供了一种将路由器上的路径频率分布转换成等价二叉树表示法的方法.

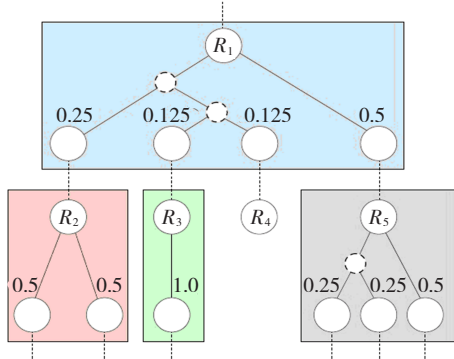


图2 模块化路径频率树

攻击树中的每个中间路由器使用路径频率计数器测量从它的直接孩子中接收到的数据包数量. 计数器的总数量仅线性依赖于路由器的度, 而与攻击子树中的路径总数量无关. 频率分布的周期快照是通过板载哈夫曼编码器生成对应的编码频率哈夫曼二叉树. 最后在攻击路径树中, 连接路由器与其直接孩子的边被这些时变哈夫曼树 (H_{R_i}) 代替, 而获得路径频率树. 图1中攻击树经过频率嵌入后得到的路径频率树如图2所示.

式(10)表示个构造路径频率树的自下而上的方法的递推关系:

$$T_{R1} = H_{R1} \cup T_{R2} \cup T_{R3} \cup T_{R4} \cup T_{R5} \quad (10)$$

2.2 物理表示

在每次迭代中, 哈夫曼编码的顺序相对于路由器直接子树的最初顺序可能发生改变, 如图1和图2中路由器 R_5 的子树顺序. 因此攻击路径频率树的物理表示应该能够正确地将不同子攻击子树以及计算后的嵌入频率和它在攻击路径树中的直接孩子相关联. 我们使用平衡括号法来优化实现严格排序和压缩表示.

平衡圆括号法一个最优的树表示方法, 只需 $2m$ 比特即可简洁地表示出 m 节点的树结构^[13]. 该过程通过前序树遍历完成, 在初次访问节点时会生成 “(”, 访问完该节点的子树后, 又生成 “)”. 可以分别使用比特 0 和 1 替换掉两个圆括号, 压缩表示 $2m$ -bit 的树结构. 至于有节点标识的树, 同样对节点标记进行前序遍历, 用 $m \log(m)$ 比特表示. 这样表示节点标记二叉树的理论下界约为 $m(2 + \log(m))$ 比特.

节点标记平衡圆括号表示法可以确保网络中的

每个中间路由器的攻击子树正确顺序和关联. 路由器的 R_i 的攻击子树的物理表示如下式(11)所示, λ 表示由于抽象化而无法显示的递归扩展.

$$\begin{aligned} TR1 = & ((((((())))))) 1234 \\ & \cup (((()))) 12 \cup \lambda \cup \lambda \\ & \cup ((())) 1 \cup \lambda \\ & \cup () \\ & \cup ((((())))) 132 \cup \lambda \cup \lambda \cup \lambda \end{aligned} \quad (11)$$

攻击路径频率树的最大尺寸如式(12)所示, 式(13)给出了在每一个中间路由器以简洁的递归关系自底向上构建攻击路径频率树的通用表示, 该机制中树修剪技术是可选项, ΔR_i 表示 R_i 的直接子树.

$$\text{TreeSize}_{\max} = n(d(4 + \lceil \log_2(d) \rceil)) \quad (12)$$

$$T_{R_i} = X_{R_i} (H_{R_i} \cup_{R_j \in \Delta R_i} T_{R_j}) \quad (13)$$

自底向上的路径频率计算有助于在攻击树的下游路由器上推断出任何一个上游路由器上树边的频率. 这样在每个中间路由器上使用很少的路径频率计数器就可以实现有效的频率检测, 只是因为哈弗曼取整稍微增加攻击树的尺寸和推测频率的扰动.

3 数据包和路径关联

如前面所述, 攻击树采用自底而上式构建, 各路由器都只有一个子攻击树的快照, 而非树根为受害主机的整个攻击树全局视图, 因此中间路由器无法预知攻击树上的每个数据包全局路径标识. 我们提出将数据包和路径关联, 使用检查单一数据包的简单方式, 通过有效的包标记把单一路径标识符传至受害主机, 采用带内数据包标记和最低路由器储存, 确保互联网的单一数据包回溯.

3.1 递归方法

利用分布式分而治之方法的递归特性来支持网络中的单个数据包回溯. 每一个中间路由器对其攻击子树中的特定数据包只使用局部(而非整个网络)的唯一路径标识符标记, 在信息汇集阶段, 上游的路由器(母树)将带内数据包标识信息转换成它自己的唯一路径标识符的局部命名空间. 这样不同的路径标识符通过不同的路由器从攻击树的树叶处输送到受害主机时, 这些路径标识符就从不同的局部命名空间转变为全局唯一的命名空间. 每一个中转路由器都使用简单的散列查找表 (L_{R_i}), 便于每一跳数路标识符 (P_{R_i}) 命名空间的转换.

式(14)以递归关系表示了路径标识符:

$$P_{R_i} = L_{R_i}(P_{R_i}), R_j \in \Delta R_i$$

$$\text{leaf} \sim > R_j \rightarrow R_i \sim > \text{root} \quad (14)$$

假设图1中路由器 R_1 从路由器 R_3 中接收到一个带内数据标示为 k 的数据包, 路由器 R_2 的攻击子树拥有唯一路径为 t , 则路由器 R_1 的新路径标识符为 $(t+k)$. 每一个中间路由器将从每一个直接子树处接收到的数据包的初始偏移量存在一个局部的散列查找表中, 查找表的尺寸线性依赖于与路由器的度数. 当结合使用此前提到的树剪枝算法时, 我们需要提供一个间接层便于命名空间扩展. 如图1中的路由器 R_1 将 R_2 的子攻击树从 t 修剪至 t 路径, 因为低效的利用率, $(t+k)$ 将导致不必要的命名空间发散和潜在的尺寸剧增. 因此将 k 转变成 $(t'+k)$ 能够实现最优的命名空间利用率. 可以看到利用提出的递归方法完成命名空间转换能够保证准确的单一数据包与路径进行关联.

虽然完全可以省略整个攻击路径频率编码步骤, 而通过在受害主机处使用不同的路径签名的相对分布比率, 沿着攻击树每一条边推断出数据流量. 然而我们并不支持采用该种设计方法, 原因总结如下:

- 1) 在任何中间路由器对网络数据流的先行过滤会搅乱最终到达受害主机的流量分布.
- 2) 需要在多个的中间路由器上安装大量的频率计数器, 以便中间路由器能够获取带流量权重的攻击子树的完整视图.
- 3) 由于路径签名的瞬变特性, Internet 路由拓扑结构的动态变化会进一步搅乱流量分布.

3.2 实例说明

如图3中的样本树, 节点和边标记分别表示不同的中间路由器和实际的流量分布. 每个路由器基于在它直接孩子上施加的排序顺序来获取路径标识.

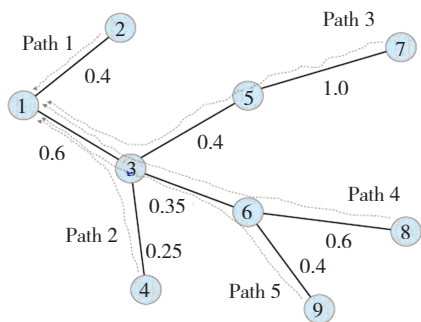


图3 带流量标注的攻击树

表1给出了每个中间路由器上的路径树和路径频率表示, 表2给出了穿越不同深度的路径2和路

径4中的带内路径标识符.

表1 路径树和路径频率树

Router	Attack Path Tree	Attack Path Frequency Tree
R_1	$2 \cup R_2 \cup R_3$	$((())) 12 \cup R_2 \cup R_3$
R_2	0	$()$
R_3	$3 \cup R_4 \cup R_5 \cup R_6$	$((())) ((())) 132 \cup R_4 \cup R_5 \cup R_6$
R_4	0	$()$
R_5	$1 \cup R_7$	$((())) 1 \cup R_7$
R_6	$2 \cup R_8 \cup R_9$	$((())) 12 \cup R_8 \cup R_9$
R_7	0	$()$
R_8	0	$()$
R_9	0	$()$

表2 带内路径标识符

Path2			Path4			
R_1	R_2	R_4	R_1	R_3	R_6	R_8
2	1	0	4	3	1	0

4 测试与分析

在现实 Internet 网络拓扑环境下对提出的方法进行了可行性和开销测试, 通过测量攻击路径树、攻击路径频率树、唯一路径识别符以及路由器查询表的尺寸大小作为评估标准来分析带外数据包标记、带内数据包标记和路由器储存. 在不同的测试场景下对每个路由器跟踪多个攻击源, 在每个树深度对 8K、32K 和 128K 这 3 个不同的树修剪值进行测试. 为从全局的角度来审视因特网中的所有的潜在受害主机, 我们假定中间路由器在任何不同树深度位置出现的概率相等.

测试分析使用到了两个真实的 Internet 拓扑结构, 数据集分别来自 CAIDA 的 Skitter^[3] 和 Lumeta 的 Internet Mapping projects 计划^[4]. Skitter 数据集在靠近受害主机相对密集, 而 Lumeta 数据集则表现出更为集中的分布.

4.1 带外流量

利用带外通道来把攻击树从不同的路由器传输到受害主机, 需要确保产生的附加流量不会在攻击过程中湮没受害主机. 图4给出了这两个拓扑结构的周期性传输的攻击路径树尺寸大小.

图5则给出了两个拓扑结构下周期性传输的攻击路径频率树的尺寸大小.

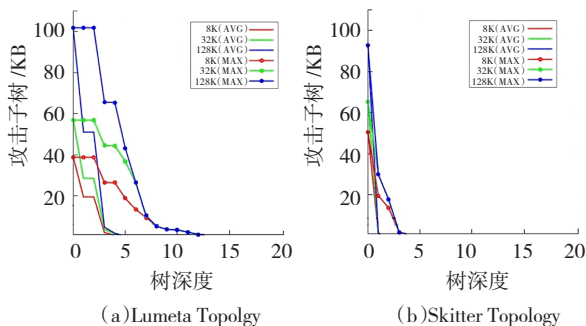


图4 带外路径树尺寸

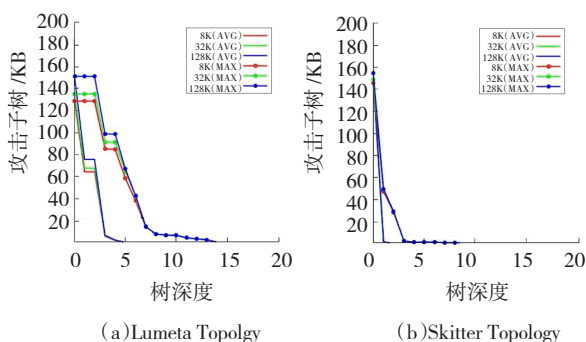


图5 带外路径频率树尺寸

通过对实验数据进行分析可以得到如下结果:

- 1) 在每个树深度的平均值曲线与这些深度的最大值曲线相似。
- 2) 存在长尾 (long tail) 现象, 表明存在大量数据包源, 且越靠近数据包源路由器在深度上越分散。
- 3) 中间路由器位置处攻击树修剪可以明显减小攻击树的尺寸。

攻击路径树和攻击路径频率树的最大尺寸分别约等于 100 KB 和 150 KB 时 (深度为 0 时达到最大值), 在任何一个路由器的整体平均值只约等于 15 KB 和 25 KB。在每个刷新间隔中, 每个路由器和每个受害主机的平均数据传输量中, 相对于 GB 单位的数据层攻击流量而言, 控制层流量的只有几十 KB。

4.2 带内流量

利用带内数据包标记建立特定数据包和路径关联。图 6 给出了两个拓扑结构的带内数据包标记尺寸。

可以看到攻击树在其深度的上 1/3 位置较为密集, 距离受害主机深度越深越分散。带内数据包标记字段为 17 bit, 足以跟踪 100 K 的数据包源, 20 bit 的则可向上跟踪上百万的源头, 这样就保证在当今 Internet 环境中真正实现单数据包溯源。

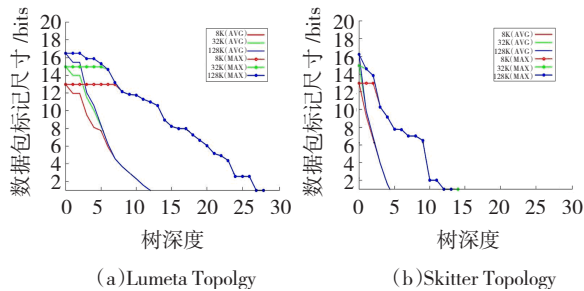


图6 带内包标记尺寸

现有的多种数据包标记方案使用不同数量的数据包, 数据包内的带内标记位数也不同, 我们采用把获取特定攻击路径检测概率的所有数据包中标记的总位数作为评测标准。图 7 比较了当路径检测概率为 50% 和 95% 时, 概率性包标记 PPM (probabilistic packet marking)、快速溯源算法 FIT (fast internet traceback)、增强概率标记 EPM (enhanced probabilistic marking)、基于 TTL 的包标记方法 TPM (TTL based packet marking)、哈夫曼方案以及我们方案的总带内数据传输量。

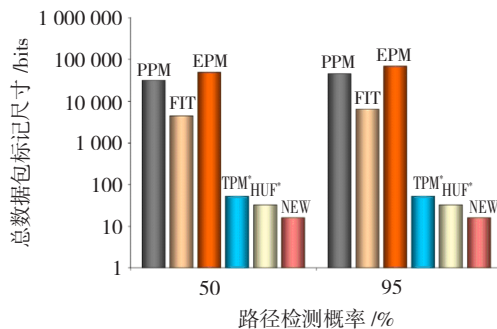


图7 网络流量开销

可见相对其他方案, 新方法不仅能够提供单独的数据包溯源保证 (无误报和漏报), 同时还能够大幅度改善流量 (包括带内和带外) 开销。

4.3 路由器存储开销

为确保特定数据包与路径关联的正确性, 在命名空间转换中使用了路由器查找散表。图 8 分别表示两种拓扑结构攻击树各深度查找表的大小。可以看到最大查找表大小约为 4 KB, 平均尺寸仅仅约为 0.4 KB。

因为所有路由器都需要在本地内存中储存查找表和攻击路径频率树, 比较本方案与其他 3 个数据包标记方法 (HASH、随机标记和采样 RMS 和哈夫曼) 的总路由器储存要求, 如图 9 所示。

从图 9 中可以看出其他方案的路由器储存要求取决于链接速度及缓存时间, 我们的方案仅取决于

同时支持的受害主机数量和各自的树剪修限制(8 K和32 K),相对于其他方案提供更好的稳定性和灵活性.

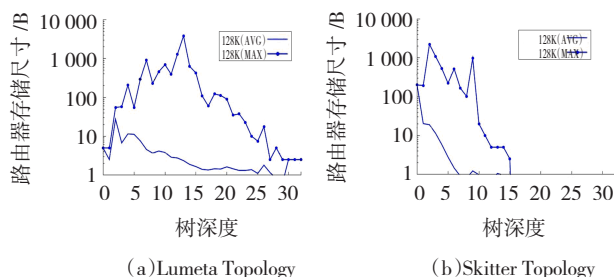


图8 查找表尺寸

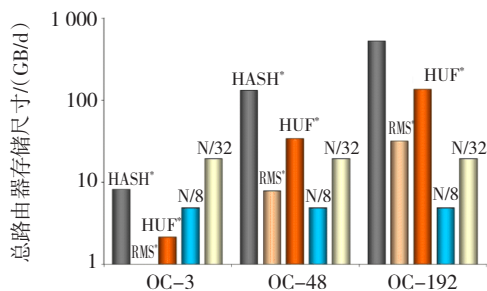


图9 路由器存储开销

5 结 语

本文针对分布式拒绝服务攻击防御中的数据标记溯源技术,提出基于分治策略用简单的递归关系分别表示攻击树构、攻击路径频率检测、数据包和路径关联这3个基本防御实现,使用单数据包带内路径标识符用于唯一的数据包和路径关联,而把攻击树构建及攻击路径频率作为独立的带外操作来处理.通过在真实网络拓扑环境下对带内流量、带外流量和存储开销的全面测试和分析,证明本方法可以高流量负荷下以很小的开销同时对大量受害主机提供单一数据包回溯保证.

参 考 文 献:

- [1] HAKEM Beitollahi, GEERT Deconinck. Analyzing Well-known

Countermeasures Against Distributed Denial of Service Attacks [J]. Computer Communications, 2012, 35(11): 1312-1332.

- [2] STEVE Mansfield-devine. DDoS: Threats and Mitigation [J]. Network Security, 2011, (12): 5-12.

- [3] LI Li, SHEN Subin. Packet Track and Traceback Mechanism Against Denial of Service Attacks [J]. Journal of China Universities of Posts and Telecommunications, 2008, 15(03): 51-58.

- [4] BELENKY Andrey, ANSARI Nirwan. On Deterministic Packet Marking [J]. Computer Networks, 2007, 51(10): 2677-2700.

- [5] LIU Jenshiuh, LEE Zhijian, CHUNG Yehching. Dynamic Probabilistic Packet Marking for Efficient IP Traceback [J]. Computer Networks, 2007, 51(03): 866-882.

- [6] LEE Fuyuan, SHIEH Shihpyng. Defending Against Spoofed DDoS Attacks with Path Fingerprint [J]. Computers & Security, 2005, 24(07): 571-586.

- [7] CHEN Weimei, CHEN Genuy. Divide-and-conquer Recurrences Associated with Generalized Heaps, Optimal Merge, and Related Structures [J]. Theoretical Computer Science, 2003, 292(03): 667-677.

- [8] WANG Xiaojing, WANG Xiaoyin. Topology-assisted Deterministic Packet Marking for IP Traceback [J]. Journal of China Universities of Posts and Telecommunications, 2010, 17(2): 116-121.

- [9] 何志宏,毛志军. 表达式与二叉树的相互转换 [J]. 电脑知识与技术, 2010, 06(05): 1201-1203.

- [10] ZHANG Changwang, CAI Zhiping, CHEN Weifeng, et al. Flow Level Detection and Filtering of Low-rate DDoS [J]. Computer Networks, 2012, 56(15): 3417-3431.

- [11] RICHARD F. Geary, NAILA Rahman, RAJEEV Raman, et al. A Simple Optimal Representation for Balanced Parentheses [J]. Theoretical Computer Science, 2006, 368(03): 231-246.

- [12] HARABURDA David, TARAU Paul. Binary Trees as a Computational Framework [J]. Computer Languages, Systems and Structures, 2013, 39(04): 163-181.

- [13] ZHANG Jun, ZHAO Hai, XU Jiuqiang, et al. Characterizing and Modeling the Internet Router-level Topology-The hierarchical Features and HIR Model [J]. Computer Communications, 2010, 33(06): 2001-2011.

- [14] SHIN' ICHI Arakawa, TETSUYA Takine. Analyzing and Modeling Router-level Internet Topology and Application to Routing Control [J]. Computer Communications, 2012, 35(08): 980-992.

(编辑: 关 毅)