软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, [doi: 10.13328/j.cnki.jos.005882]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

联盟模式下高效单包溯源方法*

鲁 宁[1,2], 张俊伟[2], 马建峰[2], 程庆丰[3], 张嘉伟[2], 王尚广[4]

[1](东北大学 计算机科学与工程学院, 辽宁 沈阳 110819)

[2](西安电子科技大学 网络与信息安全学院, 陕西 西安 710071)

[3](中国人民解放军战略支援部队信息工程大学 数学工程与先进计算国家重点实验室, 河南 郑州 450001)

[4](北京邮电大学 网络与交换技术国家重点实验室, 北京 100876)

通讯作者: 张俊伟, E-mail: jwzhang@xidian.edu.cn

摘 要: IP 协议的“无状态”特征引发了许多网络安全管理问题.为此,人们提出了单包溯源技术.然而,已有方法因激励性能低、无法增量部署、维护成本高等问题一直未被大规模推广.基于此,本文提出一种联盟模式下高效单包溯源方法,简称 TIST. 该方法首先在大规模网络上构建溯源联盟体系结构,通过剪除搭便车自治域来提高部署激励性.然后,通过融合 IP 流标记和对等过滤技术,设计一种面向溯源联盟的链路指纹建立策略,它能弱化自治域之间的溯源耦合性,实现增量部署.最后,定义一种新的面向网络前缀的计数布鲁姆过滤器,并通过优化其参数,使溯源路由器能够快速识别溯源分组,进而实现链路指纹的选择性建立,降低维护成本.通过理论分析和基于大规模真实和人工互联网拓扑的仿真实验,结果表明,相对于以往方案,TIST 在可部署性方面确实有了很大的改善.

关键词: 互联网;网络安全;IP 匿名;单包溯源;可部署性

中图法分类号: TP311

中文引用格式: 鲁 宁,张俊伟,马建峰,程庆丰,张嘉伟,王尚广. 联盟模式下高效单包溯源方法研究.软件学报. http://www.jos.org.cn/1000-9825/5882.htm

英文引用格式: Lu N, Zhang JW, Ma JF, Cheng QF, Zhang JW, Wang SG. Efficient single-packet trceback approach based on alliance theory. Ruan Jian Xue Bao/Journal of Software, (in Chinese). http://www.jos.org.cn/1000-9825/5882.htm

Efficient Single-Packet Trceback Approach Based on Alliance Theory

LU Ning[1,2], ZHANG Jun-Wei[2], MA Jian-Feng[2], CHENG Qing-Feng[3], ZHANG Jia-Wei[2], WANG Shang-Guang[4]

[1](School of Computer Science and Engineering, Northeastern University, Shenyang 110819, china)

[2](School of Cyber Engineering, Xidian University, Xi'an 710071, China)

[3](PLA Strategic Support Force Information Engineering University, State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China)

[4](State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Single-packet traceback, as a key technology to solve the network security management issues caused by the "statelessness" of IP protocol, has drawn significant attentions in recent years. However, the prior work has not been widely used due to the following disadvantages:1)inability to deploy incrementally;2)lack of deployment incentives, i.e., nondeployers can gain free riding;3) high maintenance costs. In this paper, we propose an efficient single-packet traceback approach based on alliance theory termed as TIST. It firstly establishes the traceability alliance on the large scale networks, so as to remove free-rider ASes and improve the deployment

* 基金项目: 国家自然科学基金(61601107, U1708262, 61872449);中国博士后科学基金(2019M653568);河北省自然科学基金(F2015501122, F2015501105)

Foundation item: National Natural Science Foundation of China (61601107, U1708262, 61872449); China Postdoctoral Science Foundation (2019M653568); the Natural Science Foundation of Hebei Province of China (F2015501122, F2015501105)

收稿时间: 2019-02-17; 修改时间: 2019-04-08,2019-06-14; 采用时间: 2019-07-10; jos 在线出版时间: 2020-06-08

incentives. Secondly, it designs link fingerprint establishment strategy towards traceability alliance through combining IP stream labeling and peer-to-peer filtering technics, which can weaken the traceability coupling between autonomous domains and achieve incremental deployment. Finally, it defines a novel counting Bloom Filter towards network prefixes. By optimizes its parameters, the traceable routers can quickly identify the traceable packets, and achieve the selective establishment of link fingerprints.Extensive mathematical analysis and simulations are performed to evaluate our approach. The results show that the proposed approach significantly out performs the prior approaches in terms of the deployability.

Key words: Internet, network security, IP anonymity, single-packet traceback, deployability

伴随着我国下一代互联网实施进程的推进以及新型多媒体业务的不断涌现,业务网络 IP 化已成既定事实,无论文本或是语音、视频数据都将封装在传输分组内通过 IP 网络进行转发.既然 IP 网已演变为电信运营商的基础网络平台,那么它的正常运行直接关乎国民经济和社会发展.然而 IP 协议的“无状态”特征给网络安全管理带来了如下问题:因无法真实探测 IP 网络中传输分组的路由转发情况,故难以高效地完成路径验证和网络故障诊断;因无法准确回溯路由路径和识别发送源,故在恶意匿名行为发生后难以有效地完成网络取证和安全审计.

针对上述问题,人们提出了 IP 溯源技术[1-4].它借助路由器转发 IP 分组的契机构建链路指纹,即建立传输状态,在匿名攻击或网络故障发生后运营商仅通过收集指纹分组就能重构出它们的转发路径.按照路径重构所需的指纹分组数量不同,IP 溯源方法可分为多包溯源[5-7]和单包溯源[8-12].前者将绝大部分链路指纹信息都嵌入到转发分组的头部字段中,运营商通过采集大量指纹分组才能完成路径重构;后者将链路指纹全部记录在路由器上,运营商仅需指定一个分组样本,通过查验路由器所记录的指纹信息就能识别出它的传输路径.很明显,从应用角度分析,前者更适合于追溯高速恶意匿名行为,例如拒绝服务攻击(Denial-of-Service,简称 DoS),而无法用于网络故障诊断和低速匿名行为的取证、审计.基于此,本文重点关注应用范围更广的单包溯源.

鉴于单包溯源必然会给路由器带来极大的处理开销,进而影响底层路由网络的传输性能,已有相关方法大都致力于解决如何在保证溯源质量的前提下尽可能地减少溯源路由器的开销、降低溯源系统对底层路由网络性能的影响.迄今为止,虽然已取得一些较好的成果[8-12],但是从未得到大规模部署,其中一个重要原因就是它们的可部署性差,具体表现为:1)部署意愿低.没有遵循“谁部署,谁受益”原则,要求部署域在无法获得任何收益的情况下为所有域(包括非部署域)提供溯源服务,产生严重搭便车问题.2)无法增量部署.未将各个自治域看做独立的经济或政治实体,违背平等自愿原则,要求自治域之间必须无缝协作、全网部署.3)维护成本高.要求部署域内所有溯源路由器都采用无差别的路径指纹建立方式,从而产生巨大的成本开销.

针对上述问题,本文提出一种基于联盟模式的单包溯源方法(an efficienT Single-packet Traceback based on alliance pattern,简称 TIST).首先,本方法借鉴俱乐部经济学模型,通过在大规模网络上构建溯源联盟体系结构,使得任一 Stub 域都可依据自身情况自愿地加入该联盟,并且将“溯源”功能作为一种俱乐部物品,使它能够仅在俱乐部成员之间相互提供,而俱乐部以外的成员则不再享有,从而有效地剪除了搭便车自治域,进而提高溯源方法的部署激励性.然后,通过分析溯源联盟网络下 IP 分组路由行为表现特征,遵循 TCP/IP 协议的“边缘复杂、核心简单”的原则,以弱化自治域之间溯源耦合性为目标,融合 IP 流标记和对等过滤技术,设计了一种面向溯源联盟、可增量部署的链路指纹建立策略.最后,为了维护溯源联盟成员之间这种互助关系,定义了一种空间利用率高、计算开销低的数据结构,称为面向网络前缀的计数布鲁姆过滤器,并通过优化它的参数,使溯源路由器在联盟规模动态变化的情形下也能快速、准确地从转发分组中挑选出流向其他联盟成员的溯源分组,完成有选择性的链路指纹建立.

为了验证本文提出的方法,首先对可部署性进行了理论分析,然后在基于真实拓扑的攻击场景中对其进行了实验验证,并与其它经典方法进行了对比.结果表明 TIST 除了能够增量部署,还通过改善部署激励、指纹建立、路径重构、回溯精度来提高可部署性。

本文其余部分结构如下:第 1 节对单包溯源问题进行形式化描述,进而指出本文的研究动机;第 2 节介绍本文提出的 TIST 方法,其中 2.1 节介绍整体架构,2.2、2.3 分别介绍溯源联盟网络下的链路指纹建立和溯源分组识别策略;第 3 节对 TIST 的性能进行评估,其中 3.1 节给出理论评估,3.2 节则采用实验仿真手段对分析结果进

行补充,3.3 节讨论大规模部署问题.;第 4 节介绍相关工作;第 5 节总结全文并指出下一步的工作重点.

1 问题描述和研究动机

本节首先对单包溯源问题进行形式化描述,推断出单包溯源方法应满足的性能指标,以此为参照,指出已有方法存在可部署性不足的问题,通过分析该问题产生的根本原因,进而阐明本文研究动机.

1.1 问题描述

本文将源地址伪造的IP分组称为匿名分组,其中发送和接收匿名分组的主机分别称为攻击者A和受害者V;将组成节点(路由器)由相同网络服务提供商(Internet Service Provider,简称ISP)管理、连接边由相同IGP(Interior Gateway Protocol,简称IGP)生成的网络称为自治域网络(Autonomous System,简称AS),如定义1所述;将自治域网络分为Stub和Transit两类,其中Stub域只负责转发源于自身的IP分组,而Transit域则负责为其邻居Stub域提供转发服务;将组成节点由多个自治域提供、连接边由BGP (Border Gateway Protocol,简称BGP)和IGP共同生成的网络称为底层路由网络,如定义2所述;将IGP生成的链路称为域内链路,将BGP生成的链路称为域间链路,如定义3所述;将IP包在底层路由网络的转发路径称为路由路径,如定义4所述;将用于链路指纹建立、完成溯源功能升级的路由器称为溯源路由器;将包含溯源路由器的自治域称为溯源自治域;将组成节点为溯源路由器、连接边为底层路由路径压缩生成的覆盖网络称为溯源网络,如定义5所述;将溯源网络上从V到A的路径称为攻击路径,如定义6所述.

定义 1 自治域网络可表示为一个无向图 $G_{AS}=(N_{AS},E_{AS},T_{AS})$,其中 $N_{AS}=\{n_1,.....,n_k\}$是普通路由器集合,E_{AS}是由IGP生成、连接两个普通路由器的链路集合,T_{AS}是自治域类型变量,0为Stub,1为Transit.假设顶点n_i与n_k相邻,链路(n_i,n_k)记为n_in_k.

定义 2 底层路由网络可表示为一个无向图 $G_R=(N_R,E_R,M_R)$,其中N_R是普通路由器集合,E_R是由IGP或BGP生成、连接两个路由器的链路集合,M_R是自治域集合.

定义 3 给定虚拟链路 y_iy_{i+1},如果 $y_i\in N_{ASi}$且 $y_{i+1}\notin N_{ASi}$,则 y_iy_{i+1}被称为域间链路;如果 $y_i,y_{i+1}\in N_{ASi}$,则 y_iy_{i+1}被称为域内链路.

定义 4 路由路径可表示为一个非空图 $P_R=(V_R,L_R)$,其中 $V_R=\{x_0,x_1,\ldots,x_k\}$,$V_R\subseteq N_R$;$L_R=\{x_0x_1,x_1x_2,\ldots,x_{k-1}x_k\}$,$L_R\subseteq N_R$.为了方便,可用顶点自然顺序排列表示路径,记为 $x_0\mathrm{P_R}x_k=x_0x_1\ldots x_k$,并称 $\mathrm{P_R}$ 是一条从 x_0 到 x_k 的路由路径.

性质 1 给定 $G_R=(N_R,E_R)$,$\forall x_i,x_j\in N_R$,根据路由拓扑的连通性,可推出至少存在一条路由路径 $x_i\mathrm{P_R}x_j$.

定义 5 溯源网络是由底层路由网络进行点压缩操作而获得的,也可表示为一个无向图 $G_T=(N_T,E_T,M_T)$,其中 $N_T\subseteq N_R$ 是溯源路由器集合;E_T 是连接两个溯源路由器的虚拟链路集合;$M_T\subseteq M_R$ 是溯源自治域集合.虚拟链路的生成方法如下:$\forall v_i,v_j\in N_T$,必然存在路由路径 $v_i\mathrm{P_R}v_j=v_ix_{i+1}\ldots x_{j-1}x_j$,若 $v_i\mathrm{P_R}v_j\cap N_T=\phi$,则删除路径顶点 $x_{i+1}\ldots x_{j-1}$,将 v_i,v_j 用边连接起来,生成虚拟链路 v_iv_j.

性质 2 如果 $N_T=N_R$,则 $E_T=E_R,G_T=G_R,M_T=M_R$.

定义 6 攻击路径 P_A^z是指匿名包z在溯源网络上的转发路径,表示为 $P_A^z=(V_A,L_A)$,其中 $V_A=\{y_0,y_1,\ldots,y_k\}$,$V_A\subseteq N_T$;$L_A=\{y_0y_1,y_1y_2\ldots,y_{k-1}y_k\}$,$L_A\subseteq E_T$.顶点 y_0 和 y_k 分别称为攻击入口和出口.

所谓单包溯源问题就是在溯源网络上寻找一种能为每个转发分组建立链路指纹的方法,使得网络运营商(Internet Service Provider,简称ISP)只利用单个样本分组就能够完成指纹匹配和提取,重构出整条攻击路径.很明显,除了溯源网络构建,解决单包溯源问题的另一个难点就在于链路指纹建立.至于指纹提取,它只是指纹建立的逆过程.给定虚拟链路 $\mathrm{v_iv_{i+1}}\in E_T$,IP分组是从 v_i 转发到 v_{i+1},v_{i+1} 的链路指纹建立过程可形式化为一种单射函数 $S_{vi+1}=\mathrm{f}(v_i)$,而指纹提取就是它的逆函数 $v_i=\mathrm{f}^{-1}(S_{vi+1})$,也就是说只要知道链路指纹 S_{vi+1} 就可计算出上游节点 v_i,进而推算出链路 v_iv_{i+1}.基于此,单包溯源问题就是在溯源网络上寻找一种单射函数,如定义7所述.

定义 7 $\forall$攻击路径 $P_A^z=y_0y_1\ldots y_k$,单包溯源问题就是在溯源网络上寻找一种满足单射的链路指纹建立函数f, s.t.,$\forall y_i\in P_A^z$,都有 $S_{yi-1}=\mathrm{f}(y_i)$且 $y_i=\mathrm{f}^{-1}(S_{yi-1})$,其中 y_i 为溯源路由器标识符,S_{yi-1} 为链路指纹.

如上所述,单包溯源问题的求解过程可分解为两步:构建溯源网络和寻求链路指纹函数.鉴于当前互联网的

发展状态,ISP 除了要求上述步骤必须有效和轻量,还需它们满足增量和激励部署.通过细化和量化这些要求,可推出单包溯源的性能指标应包括:

1)溯源网络构建应以激励为先,一方面严格遵守“谁部署,谁受益”原则,另一方面允许 ISP 因商业利益等原因拒绝升级,但要阻止它们搭便车.

2)溯源网络需要采用与底层路由网络相对应的多极化管理模式,通过降低 AS 之间耦合度,保证其自主性,进而实现可增量部署.

3)构建溯源网络的开销要尽可能小,以增强网络系统的可扩展性.

4)在任何情况下,都要保证链路指纹建立函数的单射性,否则就会牺牲溯源精度.

5)链路指纹建立函数和其逆函数的处理开销要尽可能小,减小溯源系统对底层路由网络的影响.

1.2 研究动机

传统单包溯源的解决思路如图 1 所示[8-12]:针对每个转发分组,它的发送域(AS_1)、经过域(AS_2 和 AS_5)和接收域(AS_6)必须都建立链路指纹,以便 ISP 反向逐跳提取指纹,重构整条攻击路径.它们存在以下部署特征:1)若想追溯 AS_1,除了 AS_1 和 AS_6,AS_2 和 AS_5 也必须按照就近原则逐次升级为部署域.只要存在一个经过域未升级,那么整个溯源任务就会失败.2)只要升级为部署域(无论 Stub AS_1 和 AS_6,还是 Transit AS_2 和 AS_5),那么它不仅需要为流入其他部署域的转发分组建立链路指纹,而且需要为流入非部署域(Stub AS_3 和 AS_4)的转发分组也建立链路指纹,从而产生巨大的成本开销.3)一旦发起溯源任务,部署域在无法收到任何收益的条件下免费为所有域(包括非部署域)提供服务.对于以盈利为目的的 ISP 来说,既然溯源升级只能无偿帮助他人,自身却无法获得任何收益,就缺乏部署意愿.部署激励通常表现在部署收益和非部署收益两方面.前者针对部署域,其部署收益越大,部署激励也越大;后者针对非部署域,若它们总能不劳而获,那么收益越大,部署激励就越小.基于此,如定义 8 所述,本文通过逐一累加所有自治域的收益来估算单包溯源方法的部署激励.以图 1 为例,已知 IS_{trc}={AS_1,AS_6,AS_2,AS_5},IS_{com}={AS_3,AS_4},且只有 stub 域之间会发送数据流.因 Transit 只是经过域,很少直接被攻击,故 $g_1(AS_1)+g_1(AS_2)+g_1(AS_4)+g_1(AS_5)=2\times2\times1/2+2\times0\times1/2=2$;因搭便车问题使得所有域都能被溯源,故 $g_2(AS_3)+g_2(AS_4)=-2\times2\times1/2=-2$;因此,IDP=0.根据性质 3,因搭便车和 Transit 部署域过多而造成已有方法的部署激励值较低.

针对上述问题,本文希望通过构建一种以 Stub 域为成员单位的溯源联盟来强化单包溯源方法的可部署性.一方面降低溯源网络的成员复杂度,仅由 Stub 域来灵活组建,这不仅有效弱化了部署域之间的溯源耦合性,实现增量部署;另一方面只为成员域建立链路指纹,既有效减少了溯源路由器的成本开销,又将非成员域从受益者列表中严格剥离,缓解了部署激励问题.

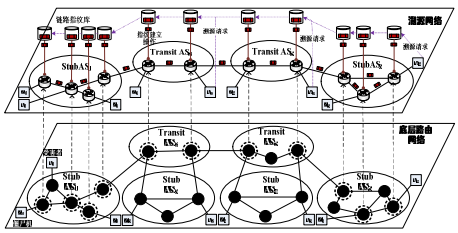

Fig.1 The basic idea in the existing single-packet IP traceback approaches.

图 1 已有单包溯源方法的基本思想.

定义 8 部署激励(Incentive for DePloyment,简称 IDP)定义为一种累加函数,可表示为 IDP=$\sum_{G_{AS}^{trc}\in IStrc} g_1(G_{AS}^{trc})+\sum_{G_{AS}^{com}\in IScom} g_2(G_{AS}^{com})$,其中 $IS_{trc}=M_T$ 表示部署溯源机制的自治域集合,$IS_{com}=M_R-M_T$ 表示未部署溯源机制自治域集合,$g_1(G_{AS}^{trc})$表示自治域 G_{AS}^{trc} 的非部署收益,$g_2(G_{AS}^{com})$表示自治域 G_{AS}^{com} 的非部署收益.$g_1(G_{AS}^{trc})=(|IS_{trc}^{stub}|/|IS_{trc}|)\times F_1^{trc}/F_1^{total}$,其中 IS_{trc}^{stub} 表示由 Stub 部署域组成的 IS_{trc} 子集,F_1^{trc} 表示向 G_{AS}^{trc} 发送匿名流但能被追溯自治域数量.$g_2(G_{AS}^{com})=-(IS_{trc}^{stub}|/|IS_{trc}|)\times F_2^{com}/F_2^{total}$,其中 F_2^{com} 表示向 G_{AS}^{com} 发送匿名流但能够被追溯自治域数量.

性质 3 在已有单包溯源方法中,搭便车和 Transit 部署域数量太多都会影响它的部署激励性.

证明:根据定义 8 可知,非部署收益越小,部署激励越大,而部署收益越大,部署激励越大.在已有单包溯源方

法中,非部署收益源于搭便车问题,它会产生大量不劳而获的非部署域;部署收益取决于部署域的数量,在匿名流总量不变的前提下,部署域越多,部署收益越小.已知部署域包括 Transit 域和 Stub 域.与后者不同,前者很少遭到直接匿名攻击,所以它的部署收益很低.因此,只要降低 Transit 部署域数量,就能提高方法的部署激励. ■

2 联盟模式下高效单包溯源方法

本文提出了一种联盟模式下高效单包溯源方法(an efficienT Single-packet Traceback based on alliance theory,简称 TIST).本节将详细阐述本方法的设计思想和具体实现细节.为此,本节组织结构如下:2.1 节主要介绍 TIST 的整体架构,之后各节分别阐述它的具体设计细节,其中 2.2 节阐述联盟网络下链路指纹建立策略; 2.3 节阐述联盟网络下溯源分组识别策略.

2.1 整体架构

TIST 的核心思想是由部署溯源机制的 Stub 域作为成员单位自愿构建一个溯源联盟,每个成员域只需承担律己的基础责任,即:联盟内每一对互为通信端的成员 Stub 域 AS_x 和 AS_y 都需为彼此提供溯源服务.当 AS_x 作为源自治域时即 IP 分组源自 AS_x 发往 AS_y,AS_x 内部的溯源路由器需要为该分组建立链路指纹,进而承担由 AS_y 发起的路径回溯任务,使它能够通过捕获异常样本分组直接完成源自治域的识别和攻击路径的回溯;同样原理,当 AS_y 作为源域、AS_x 作为目的域时,则由 AS_y 建立链路指纹建立和承担回溯任务.定义 9 对溯源联盟(Tracing Alliance,简称 TA)进行了形式化描述.很明显,这种联盟系统符合俱乐部经济学模型,将溯源功能作为俱乐部物品仅为俱乐部会员之间相互提供,而俱乐部以外的成员则不享有俱乐部的溯源服务.根据性质 4,它不仅能够有效地剪除搭便车非成员域,提高部署激励,而且能够减少成员域建立的链路指纹数量,降低处理开销.更进一步,考虑到互联网中可能同时存在多个溯源联盟,甚至它们的成员彼此交叉.当所有成员域加入同一个溯源联盟,TIST 将获得最大的激励效果,为此本文不失一般性地假设整个网络只有一个溯源联盟.按照功能,TIST 可划分为控制层面和数据层面,前者负责溯源联盟构建,后者负责 IP 分组的链路指纹建立和攻击路径回溯.接下来,我们将重点阐述这两个层面的基本实现原理.

在控制层面,TIST 支持两种联盟构建模式:永久加入和按需加入.前者允许 Stub 域以较少的准入金来加入联盟,但是必须承诺不能随意变更成员关系,否则将遭到经济惩罚.后者为了缩短成员域的溯源服务提供时间,允许 Stub 域在遭到攻击且短时间恢复无望后,以较高的准入金、临时向联盟提出加入申请.为了实现上述工作模式,溯源联盟必须具备开放性.既允许各个成员域独立部署,可依据自身实际情况(隶属关系、自身策略、网络结构和经济、政治、军事利益等)灵活加入或退出联盟;又要求成员域之间必须及时发现彼此、互相交换网络前缀.进一步结合实际网络的体系结构,溯源联盟的构建至少满足以下条件:1)灵活性.任何类型和位置的 Stub 域在任何时刻都能选择注册或退出溯源联盟.2)高效性.协同过程应简单轻权,以尽可能小的计算和通信开销来完成前缀交换,既不会降低路由收敛速度,又不给自治域之间高速通信带来明显影响.3)高可用性.随着部署规模逐渐扩展,能提供无间断服务,且不会因成员数量过大、分布广泛而降低服务质量.4)通信安全.协同过程应安全可信,防止匿名、篡改等威胁的发生.为此,本文采用客户/服务器结构模型来构建溯源联盟,将成员维护工作全部交由服务器完成,而成员域作为客户端,只需提交申请.这既简化了成员注册/退出流程,又降低了通信开销.倘若采用对等结构模型,那么成员域之间必须建立全连接通信.当联盟规模较大时,系统轻微震荡就会引发消息风暴,影响其可扩展性.

如图 2 所示, 构建溯源联盟需要由联盟注册服务器、控制服务器和溯源路由器来共同协作完成.每个成员域都配属一台控制服务器,完成以下任务:1)向注册服务器提交注册/退出请求;2)接收注册服务器向它传递的联盟成员列表信息,包括成员 AS 号和网络前缀等;3)向同层溯源路由器下发溯源验证规则,指定可为哪些 IP 分组建立链路指纹.上述环节简称“溯源验证规则部署”.整个溯源联盟配属一台注册服务器,用于动态维护溯源联盟的全局成员列表信息,管理和控制成员加盟和退出,向控制服务器实时发布联盟成员信息.上述环节简称“成员列表管理”.溯源路由器主要用于组建溯源网络、建立指纹信息和路径回溯查询.上述环节简称“溯源信息管理”.对于联盟注册服务器负载过重而引发的性能瓶颈问题和联盟开放性而引发的通信安全问题,TIST 可借鉴作者

在文献[15]提出的安全和可靠性策略.

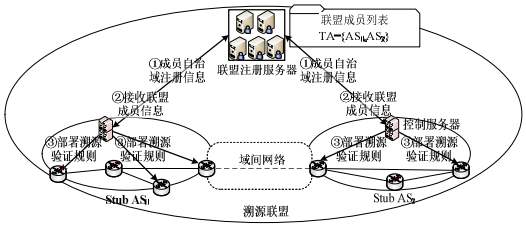

Fig.2 Tracing alliance construction in TIST (network topology from Fig.1).

图 2 TIST 的溯源联盟构建(网络拓扑源于图 1).

在数据层面,与已有方法要求 Stub 和 Transit 域全部参与,并为每个 IP 分组建立链路指纹和提供回溯任务不同,TIST 仅要求 Stub 域参与,且只为溯源联盟其他成员域建立链路指纹和提供回溯服务.基于此,在基于溯源联盟的网络体系结构中,数据通信场景被扩展为 3 类:

1)溯源联盟内单一成员域数据通信,连接同一成员域的两台主机互为通信对端,IP 分组仅在该 Stub 域内部网络中传播.在此类数据通信场景中,域内溯源路由器为每个 IP 分组一种可完成路径重构的链路指纹,同时承担由对端主机发起的路径回溯任务.

2)溯源联盟内跨成员域数据通信,隶属于不同成员域的两台主机互为通信对端,IP 分组在两个成员域内和域间网络中传播.在此类数据通信场景中,源成员域 AS_x 和目的成员域 AS_y 内不同位置溯源路由器将建立不同功能的链路指纹,其中域内溯源路由器建立面向路径重构的链路指纹,而边界溯源路由器除了这种指纹,还需建立面向源域识别的链路指纹.当 AS_y 检测出某 IP 分组 p 为攻击包之后,重构 p 在 AS_y 内的转发路径,判定 p 是否源自域外.若是,直接利用 AS_y 的边界路由器所记录链路指纹来识别出源域 AS_x,并向它发起回溯请求;反之,结束任务. AS_x 在接收请求后,重构 p 在其内部的转发路径,完成溯源任务.

3)非溯源联盟数据通信,成员域的主机与其他非联盟成员域的主机互为通信对端,IP 分组在成员域、域间和非成员域网络中传播.在此类数据通信场景中,源成员域内溯源路由器无需为 IP 分组建立任何链路指纹,也不提供任何的路径回溯服务.

到目前为止,只是介绍 TIST 的基本思想,尚留部分实现细节未讨论:

1)链路指纹建立.如何在溯源联盟环境下高效地建立链路指纹?

2)溯源分组识别.如何在溯源路由器上快速识别溯源分组,进而能够有选择性地建立链路指纹?

定义 9 溯源联盟定义为 $TA=\{G_{stub}^{trc}\in IS_{trc}|$具备联盟内溯源关系的 Stub 部署域$\}$,也就是$\forall AS_i,AS_j\in TP$,它们之间都存在联盟内溯源关系$(AS_i,AS_j)$.给定 Stub 域 AS_i 和 AS_j,如果 AS_i 能为 AS_j 提供溯源服务当且仅当 AS_j 也能为 AS_i 提供溯源服务,那么(AS_i,AS_j)就成为联盟内溯源关系.

性质 4 给定溯源联盟 TA_i,TA_i 规模越大,它的部署激励值 IDP (TA_i)也越大.

证明: 整个证明过程划分为两个阶段:1)量化溯源联盟模式下产生的部署激励效果;2)计算和比较不同联盟规模下的部署激励值.首先,本文将匿名流定义为一个三元组 $F=(s,i,d)$,其中 s 表示发送匿名流的自治域,i_{set} 表示转发匿名流的自治域集,d 表示接收匿名流的自治域.根据溯源功能,自治域可被划分为部署溯源机制的自治域和未部署溯源机制的自治域.本文将由前者组成的集合表示为 IS_{trc},由后者组成的集合表示为 IS_{com}.已知部署激励是部署收益和非部署收益之和,给定数据流 $F_j=(s_j,i_{set},d_j)$,对于任一溯源域 t_k,回溯 F_j 所带来的收益可描述如下:1)如果 t_k 是 F_j 的转发域且 F_j 的接收域也是溯源域,那么回溯 F_j 不会得到任何激励;2)如果 t_k 是 F_j 的转发域但 F_j 的接收域不是溯源域,那么回溯 F_j 就会减小一次激励;3)如果 t_k 是 F_j 的发送域且 F_j 的接收域也是溯源域,那么回溯 F_j 就可增加一次激励;4)如果 t_k 是 F_j 的发送域但 F_j 的接收域不是溯源域,那么回溯 F_j 就会减小一次激励.基于此,本文定义一种面向溯源激励的分段函数 $Y(F_j,t_k)$如下:

$$Y\left(F_j,t_k\right)=\begin{cases}-1,\left(t_k\in IS_{trc}\right)\wedge\left(t_k\in i_{set}^{j}\right)\wedge\left(d_j\in IS_{com}\right)\\ 0,\left(t_k\in IS_{trc}\right)\wedge\left(t_k\in i_{set}^{j}\right)\wedge\left(d_j\in IS_{trc}\right)\\ 1,\left(t_k\in IS_{trc}\right)\wedge\left(t_k=s_j\right)\wedge\left(d_j\in IS_{trc}\right)\\ -1,\left(t_k\in IS_{trc}\right)\wedge\left(t_k=s_j\right)\wedge\left(d_j\in IS_{com}\right)\end{cases}$$

根据拓扑位置,自治域可被划分为 Stub 域和 Transit 域两种,本文将前者组成的集合记为 AS_{stb},而后者组成的集合记为 AS_{tst}.任给数据流 $F=(s,i_{set},d)$, $s\in AS_{stb},d\in AS_{stb},i_{set}\subset AS_{tst}$.假设 $p_{stb}^{s}\in[0,1]$表示 stub s 成为攻击源概率,$p_{tst}^{il}\in[0,1]$表示 transit i_l 成为转发域的概率,$p_{stb}^{d}\in[0,1]$表示 stub d 成为受害域的概率.鉴于随机变量 p_{stb}^{s}、p_{tst}^{il}和 p_{stb}^{d}彼此独立,可知 $F=(s,i_{set},d)$的出现概率为 $p(F)=p_{stb}^{s}\times p_{tst}^{i1}\times\cdots\times p_{tst}^{i|set|}\times p_{stb}^{d}$.此外,根据源-目的,匿名流被划分为以下四组:(溯源域-溯源域)、(溯源域-非溯源域)、(非溯源域-溯源域)、(非溯源域-非溯源域),结合定义 8,整体部署激励期望的计算公式如下:$IDP=\sum_{s,i,d\in IS_{trc}}\{p[(s,i_{set},d)]\times[Y((s,i_{set},d),s)+|i_{set}|\times Y((s,i_{set},d),i)]\}+\sum_{s,i\in IS_{trc},d\in IS_{com}}\{p[(s,i_{set},d)]\times[Y((s,i_{set},d),s)+|i_{set}|\times Y((s,i_{set},d),i)]\}+\sum_{i\in IS_{trc},s,d\in IS_{com}}\{p[(s,i_{set},d)]\times|i_{set}|\times[Y((s,i_{set},d),i)]\}+\sum_{i,d\in IS_{trc},s\in IS_{com}}\{p[(s,i_{set},d)]\times|i_{set}|\times[Y((s,i_{set},d),i)]\}$.该公式由 4 部分组成:第一部分 IDP[1]代表部署收益;第二部分 IDP[2]代表非部署收益;第三部分 IDP[3]代表非部署收益;第四部分 IDP[4]代表零收益.将 Y 函数代入上述公式,进一步将 IDP 公式简化为:

$$\mathrm{IDP}=\overbrace{\sum_{s,d\in IS_{trc},i_{set}\subset IS_{trc}}P\left[(s,i_{set},d)\right]}^{IDP[1]:\text{源部署收益}}-\overbrace{\sum_{s,d\in IS_{trc},i_{set}\subset IS_{trc},d\in IS_{com}}P\left[(s,i_{set},d)\right]\times(|i_{set}|+1)}^{IDP[2]:\text{源非部署收益+中间非部署收益}}-\overbrace{\sum_{i_{set}\subset IS_{trc},s,d\in IS_{com}}P\left[(s,i_{set},d)\right]\times|i_{set}|}^{IDP[3]:\text{中间非部署收益}}.$$

鉴于 TIST 分别实现了松耦合溯源和联盟内溯源,前者使得整个溯源任务无需 Transit 域参与就能完成,因而不会产生中间非部署收益,而后者使得源溯源域不会对非联盟成员提供溯源服务,因而不会产生源非部署收益.由此可得,$\mathrm{IDP}_{\mathrm{TIST}}=\mathrm{IDP}[1]$.

然后,任给 $TA_i\subset TA_j$,必然$\exists G_{AS}^{trc}\in TA_j$ 且 $G_{AS}^{trc}\notin TA_i$,使得 TA_j 成员域能接受 G_{AS}^{trc} 所提供的溯源服务,而 TA_j 不可以,根据公式 $\mathrm{IDP}_{\mathrm{TIST}}$,$TA_j$ 的部署收益要大于 TA_i.进一步考虑到溯源联盟不含 Transit 域,给定自治域网络 G_{AS},只要在 $TA=\{AS_i|AS_1\in M_R$,且 $T_{ASi}=\mathrm{Stub}\}$情况下,任何匿名流都可被追溯,此时的部署激励值达到最大. ■

2.2 融合IP流标记和对等过滤的链路指纹建立策略

在溯源联盟环境下,为了兼顾路由网络的性能需求,TIST 的链路指纹建立应满足以下条件:1)松耦合.溯源系统不应降低底层网络各自治域的自主性,尽量保证它们在溯源过程中的松耦合.2)动态扩展.溯源网络应能动态感知新增或失效的溯源路由器,防止因节点失效而引发系统瘫痪.3)低开销.无论构建溯源网络,还是建立链路指纹,都需溯源路由器参与完成.因此,一旦它们产生大量处理开销(包括存储、计算、通信),势必影响底层路由网络转发性能.4)高精度.因溯源是为攻击过滤做准备,故溯源精度过低,势必造成过滤误报和漏报.为此,本文提出一种融合 IP 流标记和对等过滤的链路指纹建立策略.

在 2.1 节中,TIST 将路径回溯过程划分为域间和域内两阶段,前者负责攻击域识别且该域必须是溯源联盟成员,后者负责攻击域内入口路由器识别(借鉴作者之前的研究[12]).在域间回溯阶段,它要求跨过 Transit 域、直接完成攻击域识别工作,因此它只能抛弃传统的事前记录、事后取证模式.基于此,若能构建一种由成员域组成的可信网络,那么仅凭 IP 包源地址就能完成攻击域识别工作.受此启发,本文利用源地址与发送域网络前缀的隶属关系,引入轻量、已商业化的对等过滤(Mutual Egress Filtering,简称 MEF)技术,结合单向哈希密钥链,通过构建一种可信自治域网络来完成域间回溯工作.在域内回溯阶段,TIST 要求溯源路由器只为成员域建立链路指纹.为了满足要求,一种直接的方法是每个溯源路由器在建立指纹之前,都执行溯源分组识别操作,然而这必然带来巨大开销.可行的方法应该是仅由入口路由器执行溯源分组识别操作,而后续溯源路由器可根据包头属性直接识别.不过,这要求它们建立面向 IP 流的链路指纹(即溯源路由器对同一溯源路径上传播的 IP 包都执行相同的操作),而不是传统的面向 IP 分组的链路指纹.为此,本文利用路由路径无法伪造的特点,借鉴我们之前的研究[10],通过基于 IP 流标记的链路指纹建立来完成攻击域内路径重构.

本文以举例方式来阐述融合 IP 流标记和对等过滤的链路指纹建立过程.如图 3 所示,溯源联盟 $TA=\{AS_1,$

AS_6},主机 V_4 是受害者,攻击者 Z_1、Z_2 和 Z_4 同时向 V_4 发起匿名攻击;Stub AS_1 和 Stub AS_6 分别绑定网络前缀 198.1.1.0/23 和 198.3.0.0/18.如上所述,域内网络使用基于 IP 流标记的链路指纹建立,域间网络使用基于对等过滤的链路指纹建立.接下来,我们分别阐述它们的基本原理:

1)当 IP 分组 p 到达溯源网络 AS_1 的入口路由器,该路由器首先通过面向 AS_6 的分组识别器对 p 进行溯源分组判定,依据判定结果,设置溯源标记位 TF.本文通过重载 IPv4 协议闲置的预留位来充当溯源标记位.如果 p 属于溯源分组,那么将 TF 设置为 1,且为 p 建立链路指纹;反之,则将 TF 设置为 0.后续溯源路由器通过提取 TF 值来判定是否为 p 建立链路指纹.如果 TF=1,则建立;反之,直接转发.这种方法有效避免了过量建立链路指纹和重复识别溯源分组而带来的庞大开销.若 AS_6 检测出 p 为攻击包,首先通过域间回溯(具体过程在下一段描述)判定攻击域为 AS_1,然后向 AS_1 发起溯源请求.AS_1 接收请求后,首先判断该请求是否来自成员域 AS_6,若是,就向域内溯源路由器提取链路指纹,重构攻击路径;反之,拒绝溯源请求.

2)当 IP 分组 p 到达溯源网络 AS_1 的边界路由器,该路由器将 p 的源地址、目的地址与边界路由器所配置的对等过滤规则按顺序一一匹配,依据匹配结果,判定之后的过滤操作.例如,已知边界路由器 R_1 所配置的面向联盟成员的对等过滤规则为:1)permit 198.3.0.0/18 any;2)deny any 198.1.0.0/24;3)permit any any.p 的规则匹配过程可描述为:1)如果 p 的源地址包含于网络前缀 198.3.0.0/18,那么直接转发,否则转向下一条规则;2)如果 p 的目的地址包含于网络前缀 198.1.0.0/ 24,那么直接过滤,否则转向下一条规则;3)直接将 p 进行转发.上述过程可简述为:R_1 只过滤源地址不属于 AS_1 前缀、目的地址属于成员域 AS_2 前缀的 IP 分组,而将剩余分组(无论真实,还是匿名)直接转发.若 AS_6 检测出 p 为攻击包,首先判断 p 内源地址是否包含于成员域 AS_1.若是,向成员域 AS_1 发起回溯请求;反之,溯源任务停止.

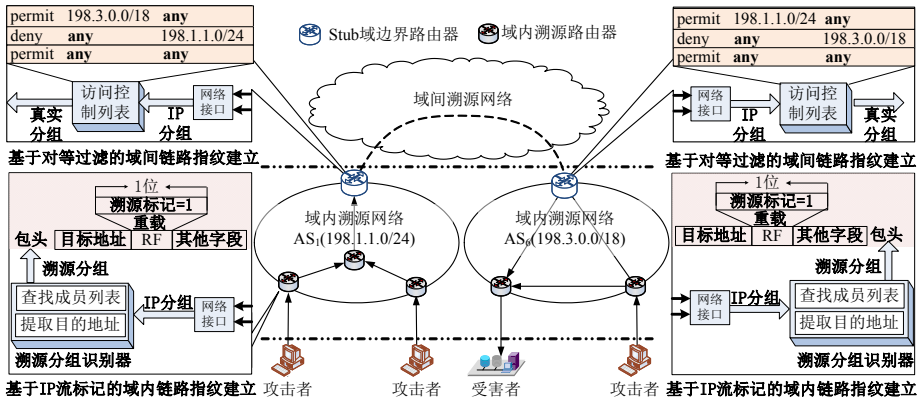

Fig.3 Link fingerprint establishment using IP stream marking and mutual egress filtering (network from Fig.2).

图 3 融合 IP 流标记和对等过滤的链路指纹建立(溯源网络源于图 2).

2.2.1 基于 IP 流标记的域内链路指纹建立

域内攻击路径难以回溯的主要原因在于传统网络的无状态性,因此若想重构路径,就需要在溯源路由器上记录少量指纹信息.进一步分析,鉴于溯源路径是由虚拟链路按序连接而成,若能为每个虚拟链路进行编码,那么只需上游溯源节点将下游链路编码写入转发分组的包头字段,下游溯源节点把该分组所携带信息看作链路指纹进行记录,就能完成反向虚拟链路识别工作.然而,同一链路可能承载多条 IP 流(本文将在同一溯源路径上传播的所有 IP 分组称为 IP 流),仅记录链路编码,不记录流特征,将无法完成面向 IP 流的链路溯源.为此,引入 MPLS 网络的标签概念,要求溯源路由器为同宿但不同源的 IP 流独立分配不同标签,用 [标签+目的地址]的 IP 流特征组合代替传统的[源地址+目的地址]组合,也就是说,对于某一溯源路由器来说,它的 IP 到达流要么携带不同的目的地址,要么携带不同的入标签.针对目的地址不同的 IP 流,该溯源路由器可分配任意出标签(相同亦可);针对目的地址相同但入标签不同的 IP 流,该溯源路由器必须分配不同的出标签,其中入标签是指 IP 到达流在进入该溯源路由器时携带的标签(即上游写入的标签),出标签是指该溯源路由器为 IP 到达流重新写入的标

签.基于此,上游溯源节点若能将下游链路编码和分配的出标签写入分组包头,下游溯源节点则通过同时记录该分组的携带信息以及新分配的出标签,建立基于 IP 流标记的溯源链路指纹.在反向回溯中,下游溯源节点只需使用出标签和目的地址就匹配出上游链路编号和入标签,前者用来判定上游节点,后者用作下一条链路回溯,迭代完成面向 IP 流的溯源链路重构任务.

我们以图 4 为例来阐述基于 IP 流标记的域内溯源的基本思想.已知图 3 包含两个成员域 AS_1 和 AS_6,不失一般性,本文只介绍 AS_1 内链路指纹建立过程和路径回溯过程.假设成员域 AS_1 的溯源网络已经建立,包括溯源虚拟链路的生成和虚拟链路编号的分配.此外,重载 IP 包头的 Identification 字段充当标记域,其中前 8 位用来存放标签,后 8 位用来存放链路编号(作者在之前研究已证明上述空间足够承载标签和链路编号).受害者位于成员域 AS_6,攻击者位于成员域 AS_1 且有两个,也就存在两条攻击路径 $P_A^{z_1}=\{R_1R_3,R_3R_4\}$ 和 $P_A^{z_2}=\{R_2R_3,R_3R_4\}$.

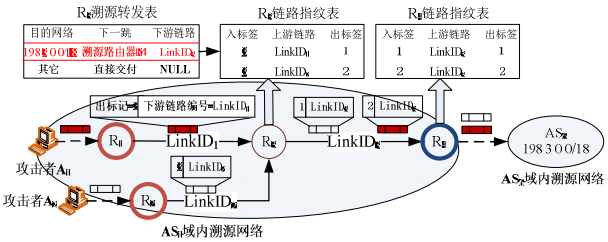

Fig.4 The intra-AS link fingerprint establishment based on IP stream marking (traceability network from Fig.3).

图 4 基于 IP 流标记的域内链路指纹建立(溯源网络源于图 3).

域内链路指纹建立过程:当攻击者 A_1 发送的 IP 分组 Z_1 到达 R_x,首先检查 Z_1 标记域,如果伪造(链路编号不正确)或为空,那么 R_x 就是入口路由器(例如 R_1 或 R_2),直接分配标准出标签 3,同时将其写入标记域,然后利用目的地址查找溯源转发表,确定下一跳溯源链路编号,并写入标记域,最后将 Z_1 转发.如果标记域不空,那么 R_x 为中间路由器(例如 R_3),提取和记录标记域信息到链路指纹表,其次按照同宿但不同源规则为 Z_1 分配出标签(即为 A_1 和 A_2 的 IP 分组分配不同标签),将其写入标记域的同时记录到链路指纹表.然后,利用目的地址查找溯源转发表,确定下一跳溯源链路编号,写入标记域.最后,将 Z_1 转发.如果标记域不空且 R_x 为边界路由器(例如 R_4),那么直接提取和记录标记域信息到链路指纹表,域内链路指纹建立过程结束,后续操作归于域间溯源.

域内攻击路径回溯过程:域间溯源能保证边界路由器 R_4 准确匹配到入标签 L_{in}.基于此,假设 AS_1 控制服务器向 R_x 提出包含(目的地址,出标签 L_{out})的溯源请求.首先查询 R_x 中包含目的地址的链路指纹表,将 L_{out} 与表条目的出标签一一匹配,查找出入标签 L_{in} 和虚拟链路编号,如果 L_{in}=3,就说明上游节点就是入口路由器,溯源请求停止;否则,就说明上游节点并非入口,需要发起新一轮溯源请求(目的地址,入标签 L_{in}).

以上内容从整体上概述了域内溯源基本原理,接下来重点从域内溯源网络构建和链路指纹建立函数两个角度展开叙述.

a. 域内溯源网络构建

域内溯源网络是以底层自治域网络为基础、溯源路由器为组成节点的覆盖网络,它的构建需要解决两个问题:1)溯源链路生成.如何在溯源网络动态伸缩条件下以尽可能低的传输成本完成溯源节点状态更新;2)链路编码分配.如何在不影响溯源精度的条件下尽可能多的压缩链路编码长度,将更多的指纹写入到标记域.

针对前者,最直接方法就是采用带外或带内传输,这不仅会带来较大的传输和计算成本,而且无法保证数据隐私安全.为此,本文使用我们之前提出的基于链路状态通告(Link Status Advertisement,简称 LSA)的域内溯源链路生成策略[12].首先通过重新定义域内路由协议——开放式最短路径优先 OSPF 的第 11 种备用 LSA,将溯源节点状态信息通过 LSA 快速传遍溯源网络,使任一溯源节点都能准确获知其他溯源节点信息.然后,压缩 OSPF 协议生成的面向全网拓扑的最短路径树,构建面向溯源网络的最短路径树,进而计算溯源转发表.最后,建立溯源邻居节点与溯源链路编号的映射表.

针对后者,最简单方法就是为每条溯源链路分配不同编码,这能保证溯源精度,但会加大编码长度.通过分

析攻击路径回溯过程,不难发现:给定任一溯源节点,只要保证与它相关的链路编码能够识别出邻居节点即可.为此,本文使用我们之前提出的基于边着色理论的链路编码策略[12],通过将溯源链路编码抽象为经典边着色问题,然后使用增量式边着色算法对它进行求解,从而将域内溯源链路的编码长度降到 8 位.

b. 域内链路指纹建立函数

在面向 IP 流的域内溯源策略中,入口路由器、中间溯源路由器和边界路由器实现不同链路指纹建立函数,分别表示为 f_1、f_2 和 f_3,其中 f_1 只为下一跳分配出标签和标记指纹;f_2 为上一跳记录链路指纹,为下一跳分配出标签和标记指纹;f_3 只为上一跳记录链路指纹.给定攻击域内溯源路径 $P_A^*=y_0y_1\ldots y_k$, $\forall y_i\in P_A^*$,f_1、f_2 和 f_3 分别定义为:

$f_1(y_0)$:= {

If(Tracing-packet) {InsertTag(3); InsertLinkID(LinkID$_{y0}$);}}

$f_2(y_{i(0<i<k)})$:= { AssignTag(y_i.Tag);

If(Tracing-packet){InsertFigItem(y_{i-1}.Dst,y_{i-1}.LinkID,y_{i-1}.Tag,y_i.Tag);InsertTag(L);InsertLinkID(LinkID$_{yi-1}$)}

$f_3(y_k)$:= {

If(Tracing-packet){InsertFingItem(y_{k-1}.Dst,y_{k-1}.LinkID,y_{k-1}.Tag);}}

其中 InsertTag 表示标签标记操作,InsertLinkID 表示链路编码标记操作,AssignTag 表示标签分配操作,InsertFingItem 表示链路指纹记录操作.

2.2.2 基于对等过滤的域间链路指纹建立

攻击域难以回溯的原因在于 Stub 域的不可信.若想直接利用源地址来识别攻击域,那么必须建立可信的域间网络.为此,引入基于 Egress Filtering 的对等过滤技术,通过在底层路由网络上构建溯源联盟来约束成员域的攻击行为.然而,鉴于溯源联盟无法囊括所有 Stub 域,也就无法阻止非成员域向成员域发送匿名流,特别是一旦这些匿名流伪装为成员域地址,就会造成攻击包中源地址再次失效,进而产生溯源误报.为此,本文在溯源联盟中引入基于密钥的身份认证机制,每个成员域都将维护一个与自己身份绑定的密钥,它们在彼此通信过程中会将密钥写入匿名包头,因非成员域无法探知密钥,也就无法伪装为成员域匿名流.在反向回溯中,受害成员首先利用匿名包的源地址来判定攻击成员,然后利用密钥来排除非成员伪造成员匿名流的情形.

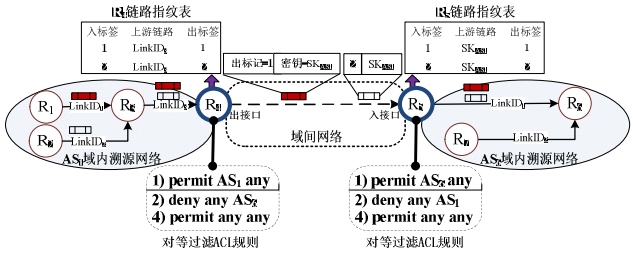

Fig.5 The inter-AS link fingerprint establishment based on MEF (traceability network from Fig.3 and Fig.4).

图 5 基于对等过滤的域间链路指纹建立(溯源网络源于图 3 和图 4).

我们以图 5 为例来阐述基于对等过滤的域间链路指纹建立基本思想.已知图 3 和图 4 的溯源联盟包含两个成员域,其中 AS_1 为攻击域,内含攻击者 A_1、A_5 和边界路由器 R_4,AS_6 为受害域,内含受害者 V_1 和边界路由器 R_5.此外,AS_1 和 AS_6 都已部署域内溯源机制,且 R_4 已为攻击流 A_1 和 A_2 分配出标签 1 和 2,并将它们写入到匿名包的标记域,也就是标记域还剩 8 位未被使用.假设 AS_1 的身份密钥为 SK_{AS1},且将 SK_{AS1} 发布于联盟注册服务器.

域间链路指纹建立过程:当攻击流 A_1 和 A_2 到达攻击域 AS_1 的边界路由器 R_4,根据它们源地址所属前缀,攻击流可分为域外和域内.针对前者,R_4 直接使用已配置的对等过滤规则来阻止它们流入域间网络;针对后者,R_4 通过在标记域中写入密钥 SK_{AS1} 来为它们重建 AS_1 域标识.当攻击流到达受害域 AS_6 的边界路由器 R_5,R_5 根据同宿但不同源规则为 A_1 和 A_2 的 IP 分组分配不同出标签,并将其写入 IP 分组的标记域,同时将入标签、密钥 SK_{AS1} 和出标签记录到链路指纹表.之后,域内溯源路由器(R_6 和 R_7)执行 2.2.1 节的域内指纹建立函数.

域间攻击路径回溯过程:受害者 V_1 携带攻击样本包 p 向 AS_6 的控制服务器发起溯源请求,该服务器收到请

求后,依据 2.2.1 节的域内攻击路径回溯过程,能够准确识别出边界路由器 R_5 以及攻击包的入标签β和密钥 SK_{AS1}.基于此,即可判定此次攻击来自域外,随后向联盟注册服务器发起认证请求.该服务器收到请求后,首先利用攻击包源地址判定攻击域位置.如果该源地址不属于任何成员域,那么此次攻击就来自非成员域(就成员域来说,它的所有域外匿名流都已被清理,只剩域内匿名流可能会流入受害域),溯源任务结束;反之,则可能来自成员域,但需要进一步证实.为此,查询密钥与成员前缀的映射表,证实 SK_{AS1} 就是成员域 AS_1 的唯一密钥.随后,携带入标签β向 AS_1 的控制服务器发起域内溯源请求.该服务器通过查询 AS_1 各边界路由器的 BGP 路由表,即可推测出 AS_1 到 AS_6 的路由路径,进而确定边界路由器 R_5.然后,再执行 2.2.1 节域内回溯过程,找到入口 R_1 和 R_2.

以上内容从整体上概述了域间溯源基本原理,接下来重点从域间溯源网络构建和链路指纹建立函数两个角度展开叙述.

a. 域间溯源网络构建

域间溯源网络是以溯源联盟为基础、成员域为组成节点的覆盖网络,它的构建需要解决两个问题:1)成员域对等过滤规则压缩.为了应对数量激增的联盟成员,如何在尽可能不损害激励性能的条件下压缩对等过滤规则;2)成员域密钥更新.为了提高系统鲁棒性,如何以尽可能小的通信开销来完成密钥更新,避免非成员域窃取密钥,进而伪造成员域的域内匿名流.

随着联盟规模的日益增大,面向部署激励的对等过滤策略会产生大量的过滤规则.然而,短缺的ACL资源使得边界路由器无法直接容纳这些规则,只能采用网络前缀压缩技术,优化过滤规则,减少资源需求.这又会产生许多“搭便车”的未部署自治域(Free Riding Non-deployer,简称FRNd),严重损害了激励机制.鉴于溯源联盟的规定,必须严格保护联盟成员.基于此,在资源受限条件下,为了应对数量激增的联盟成员,如何压缩成员的地址前缀,在防护联盟成员的同时最小化FRNd的数量,成为决定系统激励性能的关键问题.利用ACL规则的单维度特征,本文将过滤规则优化问题归为相同前缀压缩问题.给定一组成员集BL、非成员集WL、ACL规则最大使用数F_{max}和地址权重集$W(ip\in BL\Rightarrow w_{ip}=0;ip\in WL\Rightarrow w_{ip}>0)$.为了覆盖所有BL前缀,同时最小化搭便车率,该问题形式化为:

$$\min\sum_{prefix}\left[\left(\sum_{ip\in prefix}x_{prefix}\cdot w_{ip}\right)\right],w_{ip}\in W \tag{1}$$

$$s.t.\qquad \sum_{prefix}x_{prefix}\le F_{\max} \tag{2}$$

$$\sum_{prefix:ip\in prefix}x_{prefix}=1,\quad \forall ip\in BL \tag{3}$$

其中 x_{prefix} 是决策变量,x_{prefix}=1 说明网络前缀 $prefix$ 被选中,x_{prefix}=0 说明未被选中;公式(1)表明优化目标;公式(2)是资源受限条件;公式(3)是每个成员的地址前缀有且只能被覆盖一次.清华大学的毕军和加利福尼亚大学的 F.Soldo 研究团队将上述问题已经归为多维背包问题[13-14](属于 NP-hard 范畴),并且提出了相关解决算法.但是它们存在时间开销大、求解精度低等缺陷,因此我们在文献[15]进一步提出了可增量、高精度的过滤规则优化算法 EAGLE.限于篇幅,本文就不再对算法具体设计细节展开叙述.

密钥作为成员域的身份标识,一旦被非成员域劫持,就会引发溯源误报.只有不断更新密钥才会保证安全性.然而,随着溯源联盟规模逐渐增大,密钥更新范围也越来越广,这必然产生巨大通信开销.基于此,如何设计一种低开销的密钥更新策略就成为提高系统可扩展性的关键问题.考虑到各成员域边界路由器的性能容量和负载不同,它们可承受的密钥更新周期也应不同.若由联盟注册服务器集中生成和分发密钥,为了降低服务器的管理复杂度,更新周期就只能固定,这不符合上述要求.为此,本文提出一种面向成员域的动态更新策略.具体原理是:各成员域的控制服务器独立生成基于时间分片的密钥链$\{K_0,K_1,K_2,\ldots,K_m\}$,$K_i=H(K_{i-1}),0<i<m-1,K_0=x$,其中 $H:\{0,1\}^*\rightarrow\{0,1\}^\rho$,H 是单向散列函数,ρ是定值,表示输出结果的长度,*表示任意长度,$x\in\{0,1\}$是随机输入参数.每个时间片内都会生成密钥链中一个密钥,并将它下发到所有边界路由器.各成员域依据路由器的性能容量制定时间片长度.在时间片结束后,延迟时间Δ将该密钥发布到联盟注册服务器,其中Δ至少要大于密钥发布请求在互联网上的往返时间,也可以推迟若干个时间片,等溯源任务开启后,由注册服务器主动请求成员域发布密钥.在攻

击域识别过程中,注册服务器只需下载成员域最近公开的密钥,进行匹配即可.值得注意的是:为了降低各成员域生成同一密钥的可能性,可依次推算密钥链,再进行匹配.

b. 域间链路指纹建立函数

在融合对等过滤和密钥认证的域间溯源过程中,攻击域边界路由器和受害域边界路由器实现不同链路指纹建立函数,分别表示为 h_1 和 h_2,其中 h_1 为攻击域生成密钥,并为受害域分配出标签和标记指纹;h_2 只为攻击域记录链路指纹.给定攻击域为 AS_i,受害域为 AS_j,h_1 和 h_2 分别定义为:

$h_1(AS_i)$:= { GenerateKey(AS_i.SK);

If(Tracing-packet){InsertKey(SK);}}

$h_2(AS_j)$:= {

If(Tracing-packet){InsertFingItem(AS_i.SK);}}

其中 GenerateKey 表示生成密钥操作,InsertKey 表示密钥标记操作,InsertFingItem 表示指纹记录操作.

2.3 基于布鲁姆过滤器的溯源分组识别策略

溯源路由器除了完成固有的路由转发功能,还需要识别流向成员域的 IP 分组.考虑到溯源路由器的高负载以及有限的存储、计算资源,TIST 的溯源分组识别应满足如下条件:1)快速.时间复杂度要尽可能低,避免它对路由器的转发时延产生较大影响;2)轻量.空间复杂度要尽可能低,防止它对路由器的吞吐量产生较大影响.为此,本文提出一种基于布鲁姆过滤器的溯源分组识别策略.

TIST 为了建立链路指纹,需要入口路由器从 IP 分组中识别溯源分组,然后为它们建立链路指纹.一种直接的溯源分组识别策略是利用已有的 IP 包分类算法对到达分组目的地址进行最长前缀匹配,例如基于 TCAM 的分类算法、基于二叉搜索树的分类算法等,匹配成功则说明该 IP 包会流向成员域,也就可判定它为溯源分组;反之,则不是.然而,这些算法要么需要较高构建成本,要么需要较大内存和计算开销,因此并不适合 TIST.例如,在 IPv4 网络中,二叉线索树高度可达 32 层,这就意味着最坏情况下需要访存 32 次.上述策略之所以低效是因为它们采用先分类、后识别方式.若能不经分类就完成分组识别,那么处理开销自然会降低.受此启发,本文引入布鲁姆过滤器来完成识别任务,其基本思想是利用布鲁姆过滤器空间利用率高、计算时间短等特点,将所有联盟成员的网络前缀全部压缩于该数据结构中.对于每一个 IP 到达分组,使用相同的压缩算法来计算其目的地址所对应每一种网络前缀的特征值.若该值已存于布鲁姆过滤器,就证明它是溯源分组;反之,则不是.例如,给定 k 个独立哈希函数,使用它们来计算每个成员前缀,若每个哈希值占 n bits,那么 k 个哈希值对应 2^n bits 布鲁姆过滤器的位置索引,并将这 k 个位置的值设置为 1.任给 IP 分组 p,鉴于当前网络广泛采用 CIDR 技术,覆盖 p 中目的地址的前缀最多有 32 种.对于任一前缀,都使用 k 个哈希函数来计算它在布鲁姆过滤器所处位置,同时查验其值.若 k 个值都为 1,那就证明 p 就是溯源分组.不过,若将传统布鲁姆过滤器直接应用于溯源分组识别会产生如下问题:1)联盟成员的灵活退出和加入要求布鲁姆过滤器必须能够随时添加和删除成员前缀的压缩值,然而传统结构只支持添加,不支持删除;2)鉴于覆盖 IP 到达分组目的地址的网络前缀类型较多,入口路由器的高负载要求布鲁姆过滤器必须能够并行查验所有可能的覆盖前缀,然而传统结构只支持串行查验,效率较低.为此,本文设计一种面向网络前缀的计数布鲁姆过滤器(Counting Bloom Filtering toward Prefixes,简称 PCBF),如定义 10 所述,它能够在随意增添和删除成员前缀的前提下快速、准确地完成溯源分组识别.

如图 6 所示,基于布鲁姆过滤器的溯源分组识别策略的基本原理描述为:给定 PCBF=(P,E,H,S,O),其中 P={198.3.0.0/18,198.1.1.0/24,…},E={198.3.13. 28},H={H_1,H_2,…,H_k},S=ϕ,O={Insert,Delete,Find}.溯源路由器首先依据 P 中元素的掩码长度来构建计数布鲁姆过滤器,得到 S={s(/18),s(/21),s(/24),…}.一般情况下,s 的每一维计数器都由 4bit 组成,初始值为 0.然后,将 P 中所有元素插入到对应布鲁姆过滤器, 也就是$\forall p_i \in P$,执行 Insert p_i.如果成员 AS_j 退出联盟,那么 P=P-{p_j},其中 p_j 是 AS_i 的网络前缀.同时,执行 Delete p_i.最后,依据 S 中布鲁姆过滤器的掩码长度裁剪 E 中元素 e_i,进而完成查找任务,也就是$\forall e_i \in E$,Find e_i.若能返回 True,就说明到达分组是溯源分组;反之,则不是.从时间开销的角度分析,与传统布鲁姆过滤器相比,PCBF 能够完成不同长度网络前缀的并行查找,从而将查询复杂度降到 $o(k)$.

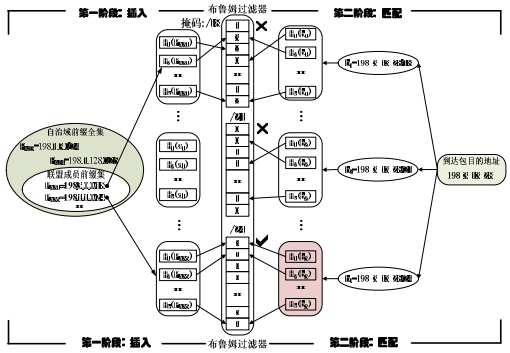

Fig.6 Traceability packet classification based on Bloom filter (from Fig.1 and Fig.3).

图 6 基于布鲁姆过滤器的溯源分组识别(源于图 1 和图 3).

定义 10 面向网络前缀的计数布鲁姆过滤器可被定义为五元组 PCBF=(P,E,H,S,O),具体含义如下:

- P={p_1,p_2,…,p_m}为所涉及的联盟成员前缀集.
- E={e_1,e_2,…,e_n}为所涉及的到达分组的目的地址集.
- H={h_1,h_2,…,h_k}为所基于的散列函数集.
- S={s_1,s_2,…,$s_{|\Pi(P)|}$}为所构建的布鲁姆过滤器集,Π表示从网络前缀到掩码长度的单向映射函数, Π(P)表示关于 P 中元素的掩码长度集,|Π(P)|为Π(P)的基数,s_i 可描述为一个长度为 x 的比特向量,s_i=(s_i^1,s_i^2,…,s_i^x),同一布鲁姆过滤器只接受掩码长度相同的网络前缀插入,因此|S|=|Π(P)|.
- O={o_1,o_2,…,o_l}为 PCBF 上关于 P 和 E 中元素的原子操作集,主要包括插入(Insert)、删除(Delete)和查找(Select).∀p_i∉P,“Insert p_i”执行步骤为:1)从 S 中搜索与掩码长度Π(p_i)对应布鲁姆过滤器 s_j;2)∀h_i∈H,计算前缀 P_i 在 s_j 中的散列地址 $h_1(P_i)$,$h_2(P_i)$,…,$h_k(P_i)$;3)将 s_j 的上述散列位置重新置位,即 $s_j^{h_1(P_i)}$++,$s_j^{h_2(P_i)}$++,…,$s_j^{h_k(P_i)}$++.“Delete p_i”的执行步骤与“Insert p_i”几乎相同,只是在置位时,前者变成 $s_j^{h_1(P_i)}$--. ∀e_i∈E,“Find e_i”执行步骤为:1)依据 S,构建 e_i 候选前缀集 C(e_i)={c_i|∀s_j∈S,c_i=σ(e_i,s_j)},其中σ表示利用 s_j 掩码长度来截取 e_i 网络前缀的函数;2)∀c_i∈C(e_i),∀h_i∈H,计算 c_i 的散列地址,得到 $h_1(c_i)$,$h_2(c_i)$,…,$h_k(c_i)$;3)若∃c_i,s.t.$s_j^{h_1(c_i)}$>0∧$s_j^{h_2(c_i)}$>0∧…∧$s_j^{h_k(c_i)}$>0,那么 e_i∈$_{\text{sub}}S$,返回 True;反之, e_i∉$_{\text{sub}}S$,返回 False.

定义 11 亚属于∈$_{\text{sub}}$:ip∈$_{\text{sub}}S$,当前仅当∃$prefix$∈S,s.t.,ip∈S,其中 ip 表示 IP 地址.

2.3.1 成员前缀数量优化

随着联盟扩张,成员前缀集 P={p_1,p_2,…,p_n}规模会越来越大,这就意味着需要压缩到布鲁姆过滤器的元素也越来越多,从而产生较大的查询误报率,这会使得更多的非联盟成员被误认为联盟成员,再次引发搭便车问题.基于此,若在不影响溯源对等关系的前提下优化成员前缀数量,就能有效地降低误报率.所谓成员前缀优化其实就是 IP 地址聚合问题,而后者会产生聚合误报率 g.给定成员前缀集 P 以及非成员前缀集 P^-,∀ip,ip∈$_{\text{sub}}P$⇔ip∉$_{\text{sub}}P^-$,求解能覆盖 P 所有元素但不产生任何聚合误报率的网络前缀.本文将该问题形式化为:

$$Object \qquad \sum_{prefix} g_{prefix} x_{prefix} = 0 \tag{4}$$

$$s.t. \quad \sum_{prefix.ip \in prefix} x_{prefix} = 1, \quad \forall ip \in_{sub} \mathrm{P} \tag{5}$$

$$x_{prefix} \in \{0,1\}, \quad \forall prefix \subseteq \Omega \tag{6}$$

其中,公式(4)表明优化目标;公式(5)表明每个成员前缀有且只能被选择一次;公式(6)表明决策变量的取值范围.

定义 12 决策变量 x_{prefix}∈{0,1},若 Prefix 被选中, x_{prefix}=1;反之,x_{prefix}=0.

定义 13 聚合误报率 g_{prefix}=∑$_{ip\in \text{prefix}}$(w_{ip}).w_{ip} 是一种符号函数,若 ip∈$_{\text{sub}}P$,w_{ip}=0;反之,w_{ip}>0.

定义 14 前缀压缩树:给定成员前缀集 P 和非成员前缀集 P^-,前缀压缩树可表示为 PC(P,P^-),它具备以下特征:1)完全二叉树;2)非叶子节点的左孩子链接为 0,右孩子链接为 1.

我们在前期工作中已经证明 IP 地址聚合优化问题满足最优子结构和子问题重叠两个特征,故可采用动态规划方法来求解[15].最直接的动态规划算法求解过程如下:首先聚合 P 中任意两种元素,并求得它所产生的误报率,其次挑选出所有误报率为 0 的聚合,将它的两个操作元素从 P 中删除,同时将操作结果加入 P,形成新的成员前缀集 P',然后在 P'中展开新一轮聚合误报率计算,直到新的成员前缀集中没有误报率为 0 的聚合.上述算法虽然简单有效但是不支持增量计算,考虑到 P 中元素会随着联盟成员的加入或退出而随时调整,若总是无法利用已有计算结果而重新开始一轮计算,那么整个过程会产生非常大的计算开销.为此,借鉴文献[14]提出的最长公共前缀树,本文提出一种新的数据结构——前缀压缩(Prefix Compression,简称 PC)树,以此为基础,提出可增量的成员前缀优化算法.

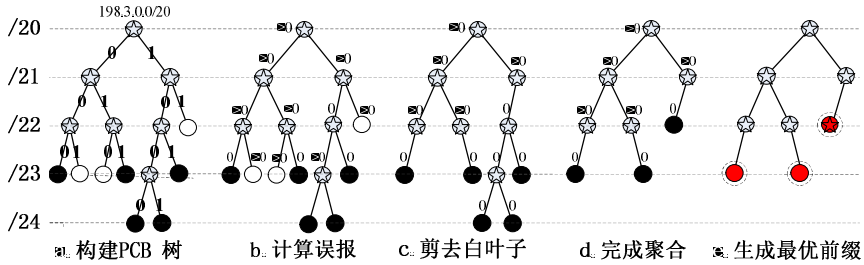

Fig.7 An example of prefix optimization procedure(From Fig.1 and Fig3). Suppose P={198.3.0.0/23,198.3.6.0/23, 198.3.10.0/23,198.3.8.0/24,198.3.9.0/24}, and P^-={198.3.2.0/23,198.3.4.0/23,198.3.12.0/22}. A black node represents a prefix of member network, a white node represents a prefix of non-member network, a star node represents the aggregation prefix, and a red node represents the prefix of selected member netowrk.

图 7.成员前缀优化过程举例(源于图 1 和图 3).假设 P={198.3.0.0/23,198.3.6.0/23,198.3.10.0/23, 198.3.8.0/24,198.3.9.0/24},P^-={198.3.2.0/23,198.3.4.0/23,198.3.12.0/22}.黑色节点表示成员网络前缀,白色节点表示非成员网络前缀,星形节点表示聚合前缀,红色节点表示优选出的成员网络前缀.

成员前缀优化过程应包含 5 步:1)构建 PC 树,其中叶子节点由 P 和 P^-元素组成,非叶子节点则由它们孩子的公共前缀组成.如图 7(a)所示,黑色节点表示 P 中成员前缀,白色节点表示 P^-中非成员前缀,星形节点表示中间聚合前缀.与已有的 LCP 树不同,PC 树的叶子节点按层次分类,只有同层次节点才可能发生聚合.2)计算每个节点 x 误报情况 $\rho(x)$,具体公式如下:

$$\rho(x)=\begin{cases} >0 & x=Leaf,\mathrm{x}\in P \\ 0 & x=Leaf,\mathrm{x}\in P^- \\ \rho_L\wedge\rho_R & \text{其他} \end{cases}. \tag{7}$$

如图 7(b)所示,黑色节点的误报 ρ_1 为 0,白色节点的误报 $\rho_2>0$.星形节点的误报 $\rho_3=\rho_L\wedge\rho_R$,其中 ρ_L 表示其左孩子误报,ρ_R 表示其右孩子误报.当 $\rho_L=0$ 且 $\rho_R>0$ 时,得 $\rho_3>0$;当 $\rho_L>0$ 且 $\rho_R=0$ 时,得 $\rho_3>0$;当 $\rho_L>0$ 且 $\rho_R>0$ 时,得 $\rho_3>0$;当 $\rho_L=0$ 且 $\rho_R=0$ 时,得 $\rho_3=0$;当 $\rho_L=\phi$,得 $\rho_3=\rho_R$;当 $\rho_R=\phi$,得 $\rho_3=\rho_L$.3)剪去 PC 树中所有白色无用节点.如图 7(c)所示,黑色叶子因参与前缀优化,故留下,而白色叶子因节点误报情况已经获得,故删除.4)执行前缀聚合,生成最优树.如图 7(d)所示,由底向上逐层删除所有父亲误报为 0 的叶子节点.5)生成最优的成员前缀.如图 7(e)所示,只需依据广度优先来遍历 PC 树中所有叶子节点,就能获得最优的成员前缀,即红色节点.

上述过程的时间复杂度分析如下:1)构建 PC 树需要执行 $|P|+|P^-|$ 次前缀插入操作,每次插入最多比较 32 次,因此它的时间复杂度上界为 $o(32\times[|P|+|P^-|])$;2)若采用分治策略来计算节点误报,通过构建递归树,已知每个节点需要计算一次,那么整体时间复杂度上界为 $o(32\times[|P|+|P^-|])$;3)只有通过遍历一次 PC 树才能剪去所有白叶子节点,它的时间复杂度上界为 $o(32\times[|P|+|P^-|])$;4)所谓聚合操作就是将误报为 0 的非叶子节点的左右孩子全部删

除,因此只需遍历一次 PC 树,时间复杂度为 $o(32\times[|P|+|P^-|])$;5)鉴于最优前缀就是叶子节点,最优前缀生成过程就是通过遍历 PC 树来输出叶子节点,它的时间复杂度为 $o(32\times[|P|+|P^-|])$.综上所述,算法的时间复杂度为各子过程之和,也就是线性阶 $o(5\times32\times[|P|+|P^-|])$.

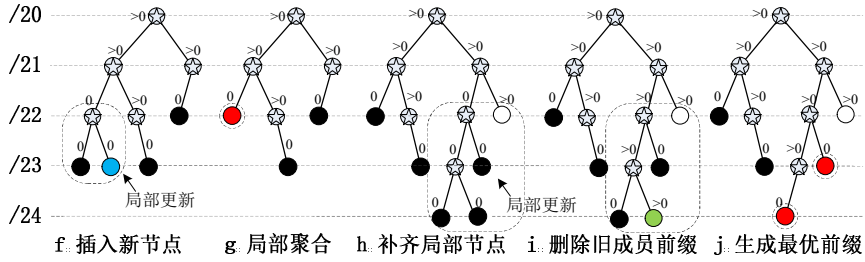

Fig.8 An example of prefix updating procedure (From Fig.7). Suppose the new added member prefix f_1=198.3.2.0, $P=f_1\cup P, P^-=P^- f_1$,the deleted member prefix f_2=198.3.9.0,$P=P-f_1, P^-=P^-\cup f_1$, a blue node represents a prefix of new added member network, and a green node represents a prefix of deleted member network.

图8.成员前缀更新过程举例(源于图7).假设新增成员前缀为f_1=198.3.2.0,$P=f_1\cup P, P^-=P^- f_1$.删除成员前缀为f_2=198.3.9.0,$P=P-f_1, P^-=P^-\cup f_1$.蓝色节点表示新增成员网络前缀,绿色节点表示删除成员网络前缀.

新增成员前缀优化过程包含 4 步:1)插入新增的成员前缀.如图 8(f)所示,蓝色节点表示新增成员,最多比较 32 次,因此时间复杂度为 $o(32)$.2)计算每个与新增成员前缀相关节点误报,相关节点数最多 32,时间复杂度为 $o(32)$.3)自顶向下逐层删除与新增成员前缀相关、误报为 0 的节点,时间复杂度为 $o(32)$.4)输出新增叶子节点,如图 8(g)所示,红色节点表示新增最优成员前缀,同时更新 P^-.综上所述,新增成员的时间复杂度为 $o(4\times32)$.

删除成员前缀优化过程包含 5 步:1)将树中与 f_1 相关节点的孩子都补全,使它们成为完全二叉节点.如图 8(h)所示,虚线方框内节点的孩子都需补齐,遍历节点最多 32 个,时间复杂度也为 $o(32)$.2)删除节点 f_1.如图 8(g)所示,绿色节点就是将要被删除节点,最多比较 32 次,因此时间复杂度为 $o(32)$.3)更新节点误报,时间复杂度为 $o(32)$.4)删除所有白色节点,时间复杂度为 $o(32)$.5)输出更新后的叶子节点,如图 8(i)所示,红色节点表示最优成员前缀,同时更新 P 和 P^-.综上所述,删除成员前缀优化的时间复杂度为 $o(5\times32)$.

2.3.2 存储空间优化

随着联盟扩张,布鲁姆过滤器数量会越来越多.鉴于存储空间是决定布鲁姆过滤器性能重要因素,而溯源路由器的内存空间是有限的,如何为不同的布鲁姆过滤器分配合理的存储容量以便它们产生尽可能小的误报率就显得非常重要.根据定义 10 可知,布鲁姆过滤器集规模取决于插入前缀的掩码长度,因此该集合中元素个数最多不超过 32.不失一般性,本文以 32 种不同掩码长度的布鲁姆过滤器为例进行讨论.此外,假设溯源路由器可用存储总量为 M,掩码长度为 i 的布鲁姆过滤器的存储容量表示为 m_i,也就是 $m_1+m_2+\ldots+m_{32}=M$.根据文献[16],单个布鲁姆过滤器的误报率 P 计算公式如下:

$$P=\left(1-e^{-\frac{nk}{m}}\right)^k, \tag{8}$$

其中 n 为插入布鲁姆过滤器的元素个数,k 为哈希函数个数,m 为该布鲁姆过滤器存储容量.很明显,P 的大小取决于 k、n 和 m.首先分析 k.本文依据路由器的计算性能和最大并行度来设置 k 的数量.然后分析 n.因误报率会随着 n 的增长而迅速变大,故本文只考虑最坏情况,即插满布鲁姆过滤器.假设某布鲁姆过滤器的掩码长度为 $I\le32$,那么它插入元素的最大数量为 2^I,所需匹配的最大 IP 地址数量为 2^{32-I+1}.已知布鲁姆过滤器 i 插满时误报率为 P_i,任给一个 IP 包,只有 32 个布鲁姆过滤器的查询结果全部准确,最终结果才会正确,PCBF 的精确度 E_{total} 为:

$$E_{total}=\prod_{i=1}^{32}(1-p_i), \tag{9}$$

基于此,本节主要研究如何最小化布鲁姆过滤器集的误报总数,也就是

$$\max\left(1-p_1\right)\times\left(1-p_2\right)\times\cdots\times\left(1-p_{32}\right), \tag{10}$$

根据定理 1 可知,只有满足 $P_1=P_2=\ldots=P_{32}$,误报总数才会最小.若想达到取等号的条件,各布鲁姆过滤器的存储容量需满足 $m_i=2^{i-1}\times m_{32}$.进一步考虑到 $m_1+m_2+\ldots+m_{32}=M$,根据等比数列求和,可得:

$$m_i=2^{31-i}\times\frac{M}{2^{32}-1}, m_{32}=\frac{M}{2^{32}-1}, \tag{11}$$

这也就是说,只要满足公式(11),即按照公式(11)为布鲁姆过滤器分配存储容量,就可获得最小误报期望.

在联盟构建初期,各个布鲁姆过滤器不可能全部插满元素,因此本文提出两种分配策略:动态分配和静态分配.前者以 m_{32} 存储空间为初始分配值,每插入一个元素就增长一个存储空间,因此其空间利用率较高,但误报率较大;后者直接以计算得出的比例依次为不同布鲁姆过滤器分配存储空间,特点是误报率低,空间利用率也低.

定理 1.当 $P_1=P_2=\ldots=P_{32}$ 时,公式(10)就会成立.

证明:已知 $1\geq P_i\geq 0$,根据算术几何均值不等式:$[(a_1+a_2+...+a_t)/n]\geq(a_1\times a_2\times...\times a_t)^{1/n}$,当前仅当 $a_1=a_2=...=a_t$ 等号成立.基于此,当$(1-P_1)=(1-P_2)=\ldots=(1-P_{32})$时,公式(10)成立. ■

3 性能评价

为了验证提出的 TIST 方法,本文进行了理论分析和仿真实验,其中 3.1 节将使用理论分析来证明方法的可部署性,3.2 节则通过基于真实网络拓扑的实验仿真来补充分析结果,3.3 节重点讨论大规模部署时方法的优势以及可能存在的问题.

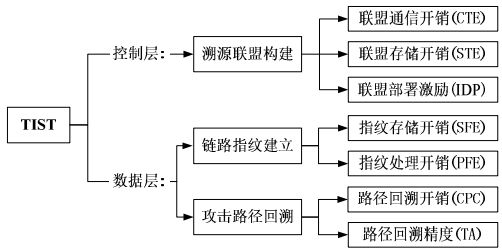

Fig.9 The performance evaluation metrics in TIST.

图 9. TIST 的性能评估指标.

3.1 理论评估

本节将使用数学方法来比较分析 TIST 与经典单包溯源方法(包括 HIT[8]、SEE[12]等)的性能指标.重点是那些 2.2 节提出的非功能、可测试指标(例如高效性、高精度),至于功能性指标(例如松耦合、动态扩展),我们在方法论述过程中就已说明.根据 2.1 节所述,TIST 方法被划分为控制层和数据层,前者承担溯源联盟构建,而后者承担链路指纹建立和攻击路径回溯.在构建溯源联盟过程中,联盟注册服务器和各成员域的控制器既需要彼此之间频繁通信,又需要各自记录成员信息.基于此,我们重点评估控制层的通信开销(Communication overhead for Tracing-alliance Establishment,简称 CTE)和存储开销(Storage overhead for Tracing-alliance Establishment,简称 STE).此外,TIST 部署激励性(Incentive for DePloyment,简称 IDP)的提高是因构建溯源联盟而造成的,因此相关评估也在这一部分.在链路指纹建立过程中,溯源路由器既需要提取指纹,也需要记录它们.故我们主要评估它的存储开销(Storage overhead for Fingerprint Establishment,简称 SFE)和处理开销(Processing overhead for Fingerprint Establishment,简称 PFE).在攻击路径回溯过程中,各成员域控制器需要提取和匹配若干溯源路由器上的指纹,而匹配又存在误报率和漏报率.因此,我们主要评估它的计算开销(Computing overhead for Path Construction,简称 CPC)和路径回溯精度(Traceback Accuracy ,简称 TA).综上所述,本节所涉及指标如图 9 所示.

3.1.1 溯源联盟构建

a. 联盟通信开销(CTE)

联盟通信开销是指在溯源联盟构建过程中所产生的数据通信量,可记为 CTE.因传统溯源方法没有实现联盟内溯源,也就无需建立溯源联盟,故 $CTE_{HIT}=CTE_{SEE}=0$.TIST 作为一种面向对等关系的单包溯源方法,要求每个成员域都必须掌握其他所有成员的前缀信息,因此每当有新成员加入或旧成员退出时,都需要联盟注册服务器将前缀变化情况下发到每个成员域的控制服务器,从而产生通信开销.给定溯源联盟 TA=(G_{stub}^{trc}),每次成员更新都会产生$|G_{stub}^{trc}|$-1 次通信开销,已知成员更新次数为 n,总的通信开销 $CTE_{TIST}=n\times(|G_{stub}^{trc}|-1)$.综上所述,$CTE_{TIST}\geq CTE_{HIT}=CTE_{SEE}$.不过,在联盟构建初期,考虑到联盟规模$|G_{stub}^{trc}|$较小,虽然成员更新频繁,但是不会产生较大通信开销.随着联盟规模增大,成员更新次数虽有降低,通信开销反而增大.

b. 联盟存储开销(STE)

联盟存储开销是指为记录联盟成员注册信息而产生的内存开销,可记为 STE.因传统溯源方法 HIT、SEE 没有建立溯源联盟,无需记录任何成员信息,因此 $STE_{HIT}=STE_{SEE}=0$.就 TIST 方法来说,它的联盟注册服务器 A 和控制服务器 B 需要收集所有成员域的前缀信息.给定溯源联盟 TA=(G_{stub}^{trc}),已知每个成员前缀需要占 5 字节,控制服务器的存储开销为 $5\times|G_{stub}^{trc}|$.经统计,全网自治域的数量约为 45k[17],也就是$|G_{stub}^{trc}|\leq 45k$,$STE_A=0.22M$ 字节.联盟注册服务器除了成员前缀,还需建立 AS 号与前缀映射表,已知 AS 号占 4 字节,映射表的存储开销为$(4+5)\times|G_{stub}^{trc}|=9\times|G_{stub}^{trc}|$.因此,联盟注册服务器的总开销为 $14\times|G_{stub}^{trc}|$字节.因此 $STE_B\approx 0.61M$ 字节.综上所述,$STE_{TIST}\geq STE_{HIT}=STE_{SEE}$.不过,就目前服务器所能提供存储能力,在可接受范围内.

c. 部属激励(IDP)

根据性质 4,$IDP_{TIST}=IDP[1]$.就 HIT、SEE 等经典方法来说,它们未考虑任何激励策略,因此 $IDP_{HIT}=IDP_{SEE}=IDP[1]-IDP[2]-IDP[3]$.鉴于 IDP[1]>0、IDP[2]>0 和 IDP[3]>0,进而推出: $IDP_{TIST}>>IDP_{HIT}=IDP_{SEE}$.需要说明的是上述推测出的 TIST 部署激励是理想值,它取决于溯源自治域需要建立严格的联盟内溯源关系.然而,受限于溯源分组识别准确度,该要求往往难以满足,进而造成准确度越低,TIST 部署激励越低;反之,则高.接下来,本节重点评估溯源分组识别准确度,记为 RAT.RAT 是指溯源路由器能准确判断 IP 到达包是溯源分组的概率,也就是识别溯源分组的误报率或漏报率.就 HIT、SEE 等经典方法来说,它们没有识别溯源分组,也无需讨论 RAT.就 TIST 来说,它采用基于 PCBF 的溯源分组识别策略,虽不会产生漏报,但会产生溯源误报,把流向非成员域的 IP 包判别为溯源分组.假设 PCBF 中存在 o 个布鲁姆过滤器,每个过滤器的误报率都相等,故记为 p,任给 IP 包 s,判断 s 为溯源分组的准确度 $RAT_{TIST}=\Pi_{i=1}^{o}(1-p)=(1-p)^o$.综上所述,已知 $p=(1-e^{-nk/m})^k$,其中 n 为插入布鲁姆过滤器的成员前缀数量,k 为哈希函数数量,m 为存储容量,鉴于 k,m 通常是定值,可推出 RAT_{TIST} 与 n 成反比.在联盟构建初期,成员前缀数量较少,RAT_{TIST} 较高,随着联盟规模增大,RAT_{TIST} 会增大.

3.1.2 链路指纹建立

a. 指纹存储开销(SFE)

为了建立链路指纹,溯源路由器单位时间内所记录的链路指纹信息数量,可记为 SFE.HIT 通过隔跳记录 IP 包来建立链路指纹.假设单位时间内到达溯源路由器 R_i 的 IP 包为 n,那么 $SFE_{HIT}\approx n/2$.SEE 通过记录 Stub 域内溯源路径片段来建立链路指纹.假设单位时间内经过 R_i 的溯源路径为 s,那么 $SFE_{SEE}\approx s$.鉴于作者在之前研究中已统计出 $s\leq 256$,而每个路径片段又与最大长度为 256 的子表外联,因此 $STE_{SEE}\leq 0.2MB$[12].与 SEE 相似,TIST 也是采用基于溯源路径的记录方式来建立链路指纹.不过, 为了实现联盟内溯源,TIST 只需要溯源路由器记录指向联盟成员的链路指纹.考虑到 Stub 域通常只设置一个流向外域的边界路由器,而且 TIST 在建立域内溯源网络时使用前缀聚合,因此 $SFE_{TIST}\leq SFE_{SEE}$.综上所述,在发生高速匿名攻击时,可推出 $SFE_{TIST}\leq SFE_{SEE}<<SFE_{HIT}$.

b. 指纹处理开销(PFE)

溯源路由器在执行链路指纹建立函数时所花费的时间,可记为 PFE.它除了影响溯源路由器的分组转发速率,还会影响网络时延.HIT 使用布鲁姆过滤器来存储 IP 包特征,当 IP 包到达后,溯源路由器 R_i 依据包头标记信息来决定是执行哈希计算,还是包标记,前者属于轻量操作,在多核条件下花费时间为 $o(1)$;相对于前者,后者更

耗时,记为 $o(s)$,也就是说 $PFE_{HIT}\le o(s)$.SEE 使用表结构来记录链路指纹信息,当 IP 包到达后,R_i 首先以包头标记信息作为匹配条件来查询溯源转发表,然后执行包标记,若使用软件来实现表结构,那么表查询开销为 $o(m/2)$,其中 m 为表项数量;若使用硬件来实现表结构,那么表查询开销为 o(1).基于此,$PFE_{SEE}\le o(m/2)+o(s)$.TIST 采用与 SEE 相似的存储结构,所不同的是 TIST 只需对溯源分组执行表查询和包标记.任给到达包,假设它成为溯源分组概率为 $p\le 1$, $PFE_{TIST}=p\times PFE_{SEE}$.综上所述,因联盟构建初期成员数量较少,故 $PFE_{HIT}\le PFE_{TIST}<<PFE_{SEE}$;随着成员增多, PFE_{TIST} 会逐渐增大,直到所有域加入联盟,$PFE_{TIST}\approx PFE_{SEE}$.

3.1.3 攻击路径回溯

a. 路径回溯开销(CPC)

TIST 为重构一条攻击路径所耗费的时间,可记为 CPC.它取决于链路指纹提取时间 TTX 和溯源路由器访问数量 n,可记为 $PC=n\times TTX$.就 HIT 来说,为了完成溯源任务,它需要沿着路由路径反向逐跳访问路由器.假设路由路径平均长度为 n,那么 $CPC_{HIT}=n\times TTX_{HIT}$.已知域间路径 n_1 和域内路径 n_2 共同组成了路由路径,即 $n=n_1+n_2$.鉴于 SEE 采用与 TIST 相似的跨域溯源架构,$CPC_{TIST}=CPC_{SEE}$.就 TIST 来说,它只需访问一次联盟注册服务器就能确定攻击域,然后回溯攻击域内路径就可识别出入口路由器.基于此,$CPC_{TIST}=n_2\times TTX_{TIST}$.链路指纹提取时间 TTX 是指溯源路由器 R_i 在收到溯源请求后,执行链路指纹提取函数所花费的时间.HIT 通过查询布鲁姆过滤器来判定 R_i 是否位于攻击路径之上.在多核条件下,$TTX_{HIT}=o(1)$;SEE 通过查询溯源转发表来确定上一跳,若该表是基于软件的,那么 $TTX_{SEE}=o(m/2)$,其中 m 为表项数量;若是基于硬件的,那么 $TTX_{SEE}=o(1)$.只要能严格禁止非成员域发起溯源请求,那么 TIST 的链路指纹提取函数与 SEE 完全相同,$TTX_{TIST}=o(m/2)$或 $o(1)$.因此,$TTX_{HIT}\le TTX_{TIST}=TTX_{SEE}$.不过,鉴于溯源任务发起不频繁,TTX 不会对路由器性能产生较大影响.也就是说,虽然 $TTE_{HIT}\le TTE_{TIST}$,但是考虑到溯源请求发送操作要远比链路指纹查询操作耗时,n 成为决定 PC 的因素.综上所述,当域内路径长度远小于路由路径时,即 $n_2<<n$,$CPC_{TIST}=CPC_{SEE}\le CPC_{HIT}$;当 $n_2=n$,$CPC_{HIT}\le CPC_{SEE}=CPC_{TIST}$.

b. 路径回溯精度(TA)

路径回溯精度是指溯源系统准确、完整回溯一条攻击路径的概率.一般来说,溯源误报率和漏报率都会影响溯源精度,前者使得系统重构出错误的攻击路径,后者使得系统重构出不完整的攻击路径.就 HIT 方法来说,它采用布鲁姆过滤器来记录压缩后的链路指纹,通常不会产生漏报率,但会生成误报率.给定攻击路径 $x_0P_Rx_k=x_0x_1\ldots x_k$,任给节点 $x_i\in P_R$,假设它的误报率为 $p_i=(1-e^{-nk/m})^k$,HIT 的溯源精度为 $TA_{HIT}=\Pi_{i=0}^{k}(1-p_i)$,其中 n 为到达 x_i 的 IP 包数量,k 为哈希函数数量,m 为 x_i 中布鲁姆过滤器的存储容量.就 SEE 方法来说,它不需要使用压缩链路指纹信息,直接进行记录,且存储开销也在可接受范围内(3.1.1 节已说明),因此既不会产生误报,也产生漏报.只是在攻击域的边界路由器上,因恶意域伪造成员域标识符而产生误报.已知标识符的总数为 2^8,假设存在 z 个恶意域,那么 $TA_{SEE}=(1-1/2^8)^z$.就 TIST 来说,它采用与 SEE 相似的域内溯源策略,因此它们的溯源精度也相同.此外,虽然 TIST 的域间溯源策略与 SEE 不同,但是它们都会因成员域标识符伪造而产生误报, 且误报率相同,即 $TA_{TIST}=TA_{SEE}$.综上所述,当发生高速匿名攻击时,因为到达溯源节点的 IP 包数量会激增,$TA_{HIT}\leqslant TA_{SEE}=TA_{TIST}$.特别是随着联盟规模增大,非成员域越来越少,$TA_{TIST}$ 会接近 1.

3.2 实验评估

3.1 节已经从理论上给出了 TIST 方法关于部属激励、系统开销的评估模型,而本小节则重点评估它在真实网络上运行时,随着模型参数调整,其性能指标的变化情况.已知 TIST 是一种融合域间过滤和域内追踪的单包溯源方法,它的性能同时受域内和域间网路结构的影响.为此,本实验搭建的溯源网络仿真平台是以互联网的域间和域内拓扑数据为基础,前者源于 UCLA 大学,后者源于 CAIDA 协会.与传统方法相比,TIST 方法的先进性在于改善了部署激励和溯源开销,而它们又与网络结构相关.基于此,本节实验首先围绕网络结构特征展开,其次分析过滤器、聚合优化、布鲁姆过滤器对部属激励性的影响,然后计算构建不同规模溯源联盟所耗费的存储和通信开销,最后评估域内溯源路由器为建立指纹和重构路径所耗费的开销.需要说明的是:传统方法无需构建溯源联盟,也不支持部署激励机制,因此相关实验无需体现它们的差别.不过,本文在系统开销方面与传统方法做了比较实验.

3.2.1 拓扑特征

根据 3.1 节,TIST 为了降耦合和提升部署激励,一方面只要求 Stub 域加入溯源联盟,另一方面只需建立面向成员前缀的追踪指纹.因此,TIST 性能与 Stub 域数量及其成员前缀存在紧密联系,本节实验围绕上述网络结构特征展开.

1)第一组实验分别统计真实网络中 Stub 域、Transit 域数量和客户-提供者关系数量,结果如图 10 所示.在真实网络中,相比于 Transit 域的 5 千多,Stub 域数量高达 3 万多,而且它们之间的 C2P 关系数量也有接近 18 万.

2)第二组实验分别统计 Stub 域和 Transit 域所拥有网络前缀数量,进而计算其补充累计分布函数,结果如图 11 所示.因为运营商对网络规划不足,所以绝大多数自治域的前缀数都较多.与 Transit 域相比, Stub 域所拥有网络前缀数量较少,80%以上都小于 2^3 个前缀.根据 3.1 推断,较少前缀有利于减少布鲁姆过滤器误报和过滤误报.

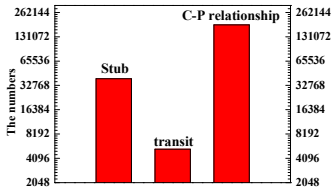

Fig.10 The network topological structure.

图 10.网络拓扑结构特征.

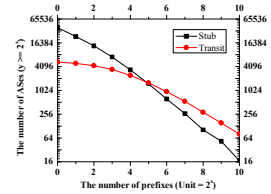

Fig.11 The number of prefixes in Stub and Transit AS.

图 11.Stub 域和 Transit 域的前缀数量.

3.2.2 溯源联盟构建

TIST 在构建溯源联盟过程中必然会产生大量存储开销和通信开销,而且规模越大,开销也会增大.为此,本实验主要关注 TIST 在真实网络上运行时相关开销的变化情况,说明它们是否成为系统大规模部署的性能瓶颈.

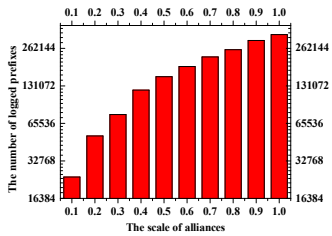

Fig.12 The recorded member prefixes on different TAs.

图 12. 不同规模溯源联盟下成员前缀记录情况.

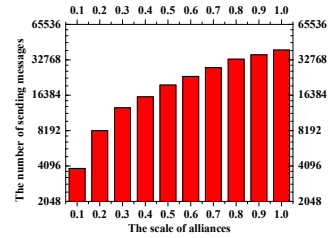

Fig.13 The communication overhead on different TAs.

图 13. 不同规模溯源联盟下通信开销情况.

1)第一组实验统计不同规模联盟下联盟注册服务器需要记录成员前缀数量,结果如图 12 所示.随着联盟规模增大,服务器记录成员前缀的数量也在增多,但是最多也不过几十 MB,对于当前处理能力较高的服务器来说,这不会成为系统的性能瓶颈.

2)第二组实验统计不同规模联盟下每增加或退出一个成员所产生的通信开销,结果如图 13 所示.随着联盟规模增大,通信开销将成线性增长,这将成为 TIST 的性能瓶颈.攻击者也可能利用该漏洞破坏溯源系统.为此,本文建议采用以下几种方法来降低开销:1)采用我们之前提出的层次化体系结构来构建溯源联盟[15];2)设计强制性的访问控制策略,严格限制一个自治域提出成员更新请求次数;3)采用延迟统一下发模式,减小下发次数.

TIST 部署激励性是由过滤器需求量、网络前缀聚合、布鲁姆过滤器所决定.为此,本实验主要关注它在不同规模的溯源联盟下过滤器需求、聚合误报和压缩误报的变化情况.为了简化实验步骤、突出比较结果,溯源联盟都是通过随机选取 Stub 域而生成.

1) 除了前缀数量,聚合误报还取决于前缀密度,密度越低,误报越大;反之,就越小.前缀密度可表示为 $\rho=S/P$, 其中 S 为成员前缀集合,P 为包含 S 的最小闭集.第一组实验统计不同规模联盟的前缀密度,结果如图 14 所示.随着联盟规模增大,成员前缀会越来越多,但是当规模达到 40%以后,前缀密度反而没有明显增长,这是因为许多

成员前缀可以无差错聚合,进而导致 P 的增长速度远没有 S 快.

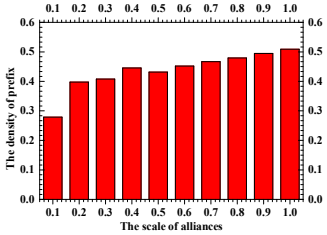

Fig.14 The number of prefixes on different TAs.

图 14. 不同规模溯源联盟下成员前缀聚合情况

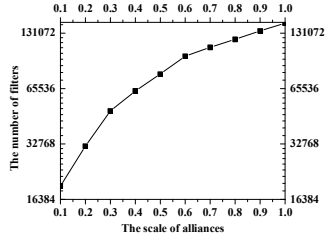

Fig.15 The number of filters on different TAs.

图 15. 不同规模溯源联盟下过滤器数量变化情况.

2) 第二组实验通过搜集不同规模联盟所需过滤器数量来得到它的补充累积分布函数,结果如图 15 所示.随着联盟规模增大,成员前缀数量也会快速增长,当全网部署时,其数可达 13 万之多.理论上来说,多一个成员前缀都需增加一个过滤器.但是随着联盟规模增大,根据图 12 所示,其聚合能力也有所增强,而过滤器数量其实是聚合后的前缀数量,因此过滤器需求量不会与联盟规模成比例增长.

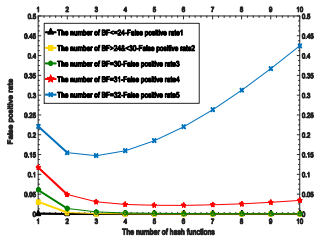

Fig.16 The false positive rate on different HASHes.

图 16. 不同数量哈希函数下误报率的变化情况.

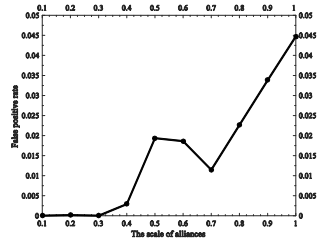

Fig.17 The false positive rate on different TAs.

图 17. 不同溯源联盟规模下误报率的变化情况.

3)第三组实验重点探讨本文提出静态分配模式下 PCBF 的最大误报率 FP_f与哈希个数 k、布鲁姆过滤器数量 x 的关系.已知 FP_f=[1-pow(e,-k×2×(2^x-1) /M)^k,其中 M 为存储容量,随着 x 增加,FP_f也越大,且 x 的取值范围为[1,32].本次实验分析当 x=1,2,…,32 时是否存在计算量和 FP_f都较小的 k,其结果如图 16 所示.当 x≤24 时,FP_f与 k 的函数关系为一条接近 x 轴的直线,因此 k 取任何值都能使得 FP_f≈0;当 24<x< 30 时,FP_f随着 k 增大而不断减少,直至 0.当 k=1 时,FP_f≈3%;当 x=30,k=1 时,FP_f≈6%,k=2 时,FP_f≈2%;当 x =31,1≤k≤6 时,FP_f随着 k 增大而不断减小,当 k>6 时, FP_f随着 k 增大而不断增大,当 k=2 时,FP_f取得最小值 2%;当 x=32,k= 3 时,虽然 FP_f的最小值可达 15%,但是在实际应用中,x=32 几乎不存在,因此用户可无需考虑.基于此,不难推出:当 x∈[1,31]时,都可取得合理 k,使得 FP_f满足用户要求.

4) 第四组实验重点探讨 PCBF 的最大误报率与联盟规模关系,结果如图 17 所示.根据统计,网络前缀长度主要集中在 11 到 32 之间,故本实验设置了 22 个布鲁姆过滤器.此外,存储容量设为 2GB.随着联盟规模增大,PCBF 误报率 FP_f总体呈上升趋势.这是因为联盟规模增大使得成员前缀数量迅速增加,进而造成插入 PCBF 元素数量 n 激增,而误报率 FP_f与 n 呈正比.不过,FP_f与联盟规模并非严格递增,例如当 x=0.5、0.6、0.7 时,FP_f反而在降低,这是因为这些规模下成员前缀聚合程度较高,使得插入数量 n 不增反减,从而导致误报率略有降低.

3.2.3 链路指纹建立

与传统方法相比,TIST 通过选择性指纹记录方式来减小存储开销.为此,本实验主要比较不同攻击规模环境下 TIST 与传统方法的溯源路由器开销情况.因为 TIST 允许各 Stub 域独立化部署和区域化管理,所管辖溯源路由器的相关开销与其他域毫无关系,所以本文随机选择了 CAIDA 数据中一个规模较大的 7018 自治域作为底层路由拓扑,以便搭建溯源网络仿真平台.

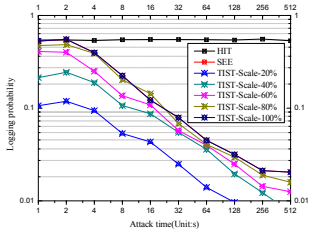

Fig.18 The fingerprint logging probability by increasing the attack time interval.

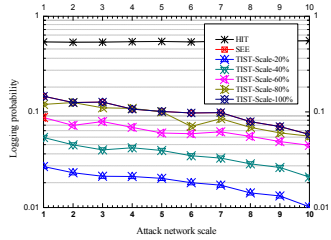

Fig.19 The fingerprint logging probability by increasing the scale of TAs.

图 18.不同攻击时段溯源路由器指纹记录比例的变化情况. 图 19.不同攻击规模下指纹记录比例的变化情况.

第一组实验搜集不同时段内溯源路由器记录指纹数量与转发分组数量的比值,结果如图 18 所示.一方面 TIST 的记录比例要远低于 HIT 的 50%,随着攻击时间逐渐增大,记录比例甚至能降低到不足 1%;另一方面随着联盟规模增大,TIST 溯源路由器所需记录的指纹数量也会逐渐增大.直到所有自治域都加入联盟,TIST 的记录比例将增加到 SEE 级别.

第二组实验搜集不同攻击规模下溯源路由器记录指纹数量与转发分组数量的比值,结果如图 19 所示.所谓攻击规模是指攻击主机占所有主机数量比例.攻击时间为 32 秒.从中可看出:1)随着攻击规模的增大,TIST 记录比例并未增长,反而降低.原因是:攻击主机增多使得网络中路由路径数减少,进而降低指纹记录数.2)随着联盟规模增大,TIST 溯源路由器所需记录的指纹数量也会逐渐增大,这与第一组实验得出相同结果.

3.2.4 攻击路径回溯

TIST 通过域间对等过滤方式来降低路径回溯开销.为此,本实验假设成员域利用密钥链技术能够准确识别彼此,主要关注不同路径长度下 TIST 与传统方法的路径回溯开销和回溯精度.实验设置与 3.2.3 节相同.

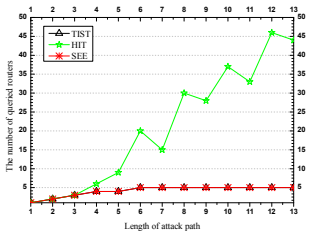

Fig.20 The number of queried routers during path reconstruction.

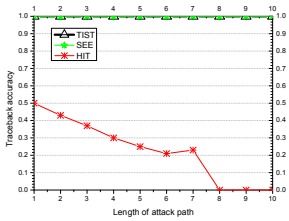

Fig.21 The traceback accuracy by increasing length of attack path.

图 20.路径重构过程中查询溯源路由器数量. 图 21.不同路径长度下溯源精度的变化情况.

第一组实验搜集不同长度路径重构过程中查询溯源路由器数量变化情况,结果如图 20 所示.从中可看出:1) TIST 查询溯源节点数量不会随着路径长度而变化,这与 SEE 方法相同;2)TIST 通常只需查找 4、5 个溯源节点就可以追溯到一个源头,而 HIT 却需要几十个.当面对成千上万的攻击源,TIST 的可扩展性更强.

第二组实验搜集不同长度路径下回溯精度变化情况,结果如图 21 所示.从中可看出:相比于 HIT,TIST 和 SEE 的回溯精度能够始终维持在较高水平,而 HIT 的回溯精度随着路径增长,会有明显降低,甚至到 0.原因是 HIT 采用了指纹压缩技术,使得误报率和漏报率随着路径增长而增大,TIST 和 SEE 无需压缩,进而准确匹配.

3.3 讨论

除了较强的部署激励性、可增量部署和较低的维护成本,TIST 方法在大规模网络部署时还具备以下特点:

1)不会分裂网络.TIST 沿用了传统溯源方法的“改良型”研究思路,尽量不对已稳定运行数十年的现有互联网协议进行改进和更新,而是利用已有协议框架以及现有技术完成 IP 溯源.在域内网络中,TIST 通过重载 IP 包头的 Identification 字段来存放 IP 流标记信息,而不是添设新的包头字段;在域间网络中,TIST 通过已广泛流行的

Egress filtering 技术来实现对等过滤,而不是采用“革新型”的源地址认证技术.基于此,它无需刻意的协议转换就能实现部署域-非部署域之间通信、跨联盟成员域之间通信,以及溯源分组在普通 Transit 域中透明传输.这也就意味着 TIST 所构建的溯源联盟并不会将路由网络割裂,更像是建于之上的层叠网络,非成员域的底层路由设备完全感觉不到它的存在,同时它也不会影响路由设备的转发操作.

2)易与 SAVI 标准整合.SAVI(Source Address Validation Improvements,简称 SAVI)是由清华大学吴建平院士团队推动 IETF 成立的工作组,致力于源地址验证技术标准化,以期建立可信任下一代互联网.目前, SAVI 已提出一种包括接入、域内、域间 3 个层次的真实 IPv6 源地址验证系统,分别在主机、IP 地址前缀和自治域粒度上保证 IP 地址的真实性.在域间网络中,它们通过构建层次化的基于标签替换的信任联盟来验证源地址真实性.TIST 的溯源联盟与 SAVI 的信任联盟并不冲突,反而可以将前者整合到后者的网络上.图 22 给出整合示例,具体操作步骤如下:a.已知信任联盟既包含 Stub 域,也包含 Transit 域,而溯源联盟的成员只能从信任联盟的 Stub 成员中选取,未加入信任联盟的 Stub 域无法加入溯源联盟.b.信任联盟是建立在 IPv6 网络上,而 TIST 构建的溯源联盟是基于 IPv4 网络.为了能将 TIST 顺利迁移到信任联盟,每个溯源联盟成员通过重载 IPv6 包头的逐跳扩展字段来存放 IP 流标记[18],进而建立 Stub 域内链路指纹.c.ISP 利用信任联盟已然建立可信域间网络的特点,不再要求溯源联盟成员的边界路由器执行对等过滤,进而建立域间链路指纹.

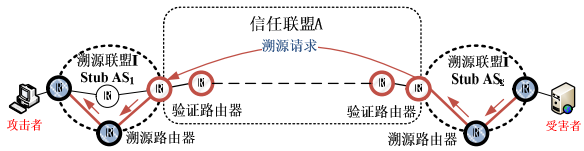

Fig.22 Integrate trust alliance A with traceability alliance I.

图 22.将信任联盟 A 与溯源联盟 I 相整合.

3)不会降低网络吞吐量.虽然 TIST 使用了包标记技术来记录链路指纹,但是并未增加 IP 包头的整体长度.因此,TIST 的指纹建立过程不会给网络带来额外的带宽开销,也就不会影响它的网络吞吐量.

当然本文提出的 TIST 方法在大规模部署时也会存在以下问题:1)鉴于 Stub 域在正常情况下都会选择安全可信的 Transit 域作为它的网络服务提供商,因此 TIST 假设溯源分组在穿越 Transit 域的过程中不会发生包标记信息被篡改的情况.然而,真实网络可能发生 BGP 路由前缀劫持攻击,使得 Stub 域在无意识的状态下被迫连接一些恶意自治域,它们通过恶意篡改溯源分组的标记信息来破坏 TIST 的回溯操作,降低它的溯源精度.2)TIST 虽然不会影响网络吞吐量,但是会增加网络延迟.溯源路由器除了转发操作,还需建立链路指纹,这会增加它的处理开销,3.1.2 节已对它进行了理论评估.若采用硬件升级,那么处理开销较小;若采用软件实现,那么开销较大.

4 相关工作

由于 IP 协议的“无状态”引发了一系列网络安全管理问题,因此 IP 溯源技术一经提出就受到人们广泛关注.按照路径重构所需指纹分组数量,IP 溯源技术可划分为基于覆盖网协同标记的多包溯源方法和基于覆盖网协同记录的单包溯源方法.接下来,本节将阐述上述方法基本思想,指出缺乏激励的协同方式是阻碍溯源系统大规模部署的重要原因.

1)基于覆盖网协同标记的多包溯源方法.其基本思想是在物理路由网络上建立由溯源路由器组成的覆盖网络,各溯源路由器利用转发 IP 分组的契机,以协同方式将自身标识信息或链路标识信息写入到 IP 分组的标记空间中,在 DoS 攻击发生后,受害者通过搜集大量携带标记信息的指纹分组,进一步汇总和分析,就可还原出完整的入侵路径或入口节点.经典方法包括概率性包标记[5]、确定性包标记[6]、动态确定包标记[7]等.它的优势在于可部署性强,既不需要高额的维护成本,也不要求 Transit 域参与.但是,存在拓扑隐私泄漏和应用范围有限(适合追溯高速匿名行为的发送源,而无法用于网络故障诊断和低速匿名行为的取证、审计)等问题.

2)基于覆盖网协同记录的单包溯源方法.其基本思想是在物理路由网络上建立溯源覆盖网络,使覆盖节点能够以协同方式来记录 IP 分组的路由转发指纹信息,从而弥补传统互联网无状态的缺陷, ISP 只需搜集覆盖网络所记录单个样本分组就可还原出整条攻击路径和入口节点.经典方法包括基于 IP 分组的单包溯源[8]、基于路由器接口的单包溯源[9]和基于数据流的单包溯源[4].与方法类别 1)相比,它的优势在于既能保护拓扑隐私,又有较广应用范围.然而,其可部署性较差,表现为:无法增量部署;部署意愿低;维护成本高.本文重点关注该类问题.

基于路径的单包溯源(包括 PSIT[10]、S3T[11]和 SEE [12])是作者早期的研究工作. PSIT 的基本思想是利用路由路径不可伪造的特点,借鉴 MPLS 网络中标签分发原理,在路由网络上建立反向路由路径指纹;在路径回溯阶段,利用 MPLS 的标签路径建立原理来还原攻击路由.与其他方法比,PSIT 具备存储开销小、路径重构时间短等优势.然而,它因采用集中式标签发布和无差别标签回收等策略而存在溯源精度问题.为此,作者提出一种改进方法,称为S3T,通过灵活配置溯源路由器的存储资源、自适应调整链路指纹的存储时间等手段来提高溯源精度. S3T 在面对大规模网络时因溯源系统结构扁平化、不支持溯源节点动态加入和颜色复用率低而产生可扩展性问题.为此,作者进一步提出一种域间过滤、域内追踪的层次化系统架构模型 SEE,同时利用边着色理论和 LSA 通告重载等手段来提高系统的可扩展性.虽然 SEE 已初步提出溯源联盟的概念,但是既没有提出联盟构建方法以及联盟模式下链路指纹建立和路径回溯策略,也没有考虑过滤器短缺和溯源分组识别等问题,因此 SEE 依然存在因搭便车、无法增量部署等原因而引发的可部署性问题.本文的 TIST 方法就是针对这些问题提出的.首先,它对溯源联盟进行形式化定义,详细地给出了它的构建策略,从而有效地剪除了搭便车自治域.然后,设计了一种溯源联盟环境下可增量部署的链路指纹建立方法,对所涉及的过滤器短缺、非成员域恶意破坏等问题都提供了解决策略.最后,提出一种高效地溯源分组识别算法,使溯源路由器能够快速识别流向成员域的 IP 分组,完成链路指纹的选择性建立,降低了维护成本.

近几年,针对单包溯源方法的可部署性差问题,有学者提出了基于云的 IP 溯源方法[4],它允许 ISP 向网络客户提供溯源增值服务,在客户遭到攻击后,只有向相关 ISP 交付一定费用后,才可获得溯源服务,进而重构攻击路径.它与本文所提 TIST 有以下不同:1)前者采用现收现付的商业模式,更多从用户角度出发来提高部署激励,需要 ISP 付出高额成本,而后者采用订阅付费的商业模式,更多从 ISP 角度出发来提高部署激励,ISP 所付成本较低;2)前者要求经过域必须参与溯源过程才能完成路径重构,溯源域之间是紧耦合关系,极大地增加系统部署难度,而后者仅需攻击域参与即可完成溯源,部署难度大大降低.

5 结论及未来研究工作

为了提高可部署性,本文提出一种基于联盟模式的单包溯源方法,简称 TIST.该方法允许各个 Stub 自治域依据自身情况灵活构建溯源联盟,以便剪除搭便车自治域,强化部署意愿,以及减少链路指纹建立数量,降低维护成本.在溯源联盟环境下,TIST 需要解决两个问题:1)如何在松耦合环境下实现单包溯源;2)如何在高速且资源受限环境下识别溯源分组.为此,本文提出了 Stub 到 Stub 的单包溯源策略和基于布鲁姆过滤器的溯源分组识别策略.通过理论分析和基于大规模真实拓扑的仿真实验结果表明,TIST 在可部署性方面有了显著提高.

到目前为止,我们还没有能力在实际网络中对提出的 TIST 方法进行部署,以评估其在仿真环境之外性能.尤其是溯源系统需要对当前路由器的软、硬件进行改造和升级,由于宏大的部署规模、网络的不稳定等因素,仿真环境必然会与实际环境存在偏差,特别是在溯源功能开启后路由器转发性能的变化情况.然后,TIST 虽然通过构建溯源联盟提高了部署激励性,但是未将自治域异构性考虑在内,使得溯源成本并不相同的自治域会获得相同收益,从而出现支出/收益不平等现象,再次抑止部署激励性.最后,TIST 虽然增强了可部署性,但是未将鲁棒性考虑在内,一旦攻击者利用僵尸机不断向控制服务器发起溯源请求,就会造成溯源系统因负载过重而宕机.因此,我们希望在今后,1)以当前本文研究为基础构建真实网络环境下的试验平台,补充 TIST 对路由器分组转发性能的影响.2)在订阅付费模式上,进一步设计一种面向服务的单包溯源方法,使得每个服务提供域能够依据溯源成本逐次向服务使用者收取费用,保证收支平衡.3)设计一种面向溯源系统的访问控制策略,阻止恶意用户随意发起溯源服务请求,增强系统鲁棒性.

References:

[1] Lu N and Hu W.H, “SAFE: a scalable filter-based packet filtering scheme,” China Communications, 2016, Vol. 13(2), pp. 163-177.[doi:10.1109/CC.2016.7405734]

[2] Lu N, Wang Y.L, Su S and Yang F.C, “Filtering location optimization for the reactive packet filtering,” Security and Communication Networks, 2014, Vol. 7(7), pp. 1150-1164.[doi:10.1002/sec.848]

[3] Lu N , Wang Y and Shi W, “Filtering Location Optimization for Defending Against Large-Scale BDoS Attacks,” Chinese Journal of Electronics, 2017, Vol.26(2), pp.435-444.[doi:10.1049/cje.2017.01.016]

[4] Cheng L, Divakaran D M, Ang W K, “FACT: A Framework for Authentication in Cloud-Based IP Traceback,” IEEE Transactions on Information Forensics & Security, 2017, 2017(99):604-616. [doi: 10.1109/TIFS.2016.2624741]

[5] Nur A Y, Tozal M E, “Record route IP traceback: Combating DoS attacks and the variants,” Computers & Security, 2018, Vol.72, pp.13-25. [doi: 10.1016/j.cose.2017.08.012]

[6] Aghaei-Foroushani V, Zincir-Heywood A N, “IP traceback through (authenticated) deterministic flow marking: an empirical evaluation,” Eurasip Journal on Information Security, 2013, 2013(1):1-24. [doi: 10.1186/1687-417X-2013-5]

[7] Yu S, Zhou W, Guo S, et al. A Feasible IP Traceback Framework through Dynamic Deterministic Packet Marking. IEEE Transactions on Computers, 2016, 65(5):1418-1427. [doi: 10.1109/TC.2015.2439287]

[8] Gong C, Sarac K. A More Practical Approach for Single-Packet IP Traceback Using Packet Logging and Marking. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(10):1310-1324. [doi:10.1109/ TPDS.2007.70817]

[9] Vijayalakshmi M, Shalinie S M and Ming-Hour Y, “HPSIPT: A high-precision single-packet IP traceback scheme,” Computer Networks, 2018, Vol.143, pp.275-288.[doi:10.1016/j.comnet.2018.07.013]

[10] Lu N, Wang Y.L, Su S, Yang F.C. A novel path-based approach for single-packet IP traceback. Security and Communication Networks, 2013, Vol.7(2), pp.309-321. [doi:10.1002/sec.741]

[11] Lu N, Wang S.G, Li Feng, Shi W.B, Yang F.C, “An efficient and precise approach for single-packet traceback,” Journal of Software, 2017, 28(10):2737-2756. [doi:10.3724/SP.J.1016.2015. 00500]

[12] Lu N, Wang S.G, Li Feng, Shi W.B, Yang F.C. A dynamically scalable and efficient approach for single-packet IP traceback. Journal of Software, 2018,29(11):3554-3574. [doi:10.13328/j.cnki. jos.005330]

[13] Liu B.Y, Bi J, Athanasios V.V. Toward incentivizing anti-spoofing deployment. IEEE Transactions on Information Forensics and Security, 2014, 9(3):436-450.[doi:10.1109/TIFS.2013.2296437]

[14] Soldo F, Argyraki K, Markopoulou A. Optimal source-based filtering of malicious traffic. IEEE/ACM Transactions on Networking, 2012, 20(2):381-395. [doi:10.1109/TNET.2011.2161615]

[15] Lu N, Li Feng, Wang S.G, Shi W.B, Yang F.C, “A hierarchical anti-spoofing alliance construction approach based on MEF,” Journal of Software, 2017. [doi:10.13328/j.cnki. jos.005517]

[16] Geravand S and Ahmadi M, “Bloom Filter Applications in Network Security: A State-of-the-Art Survey,” Computer Networks, 2013, Vol.57 (18):4047-4064. [doi: 10.1016/j.comnet.2013.09.003]

[17] Cidr Report, 2013. URL: http://www.cidr-report.org/as2.0/.

[18] Kamaldeep, Malik M, Dutta M, “Implementation of single-packet hybrid IP traceback for IPv4 and IPv6 networks, ” IET Information Security, 2018, 12(1):1-6.[doi: 10.1049/iet-ifs.2015.0483]

附中文参考文献:

[11] 鲁宁,王尚广,李峰.一种高精度、低开销的单包溯源方法.软件学报,2017,28(10):2737-2756.[doi:10.3724/SP.J.1016.2015.00500]

[12] 鲁宁,王尚广,李峰,史闻博,杨放春.可动态扩展的高效单包溯源方法.软件学报,2018,29(11):3554-3574. [doi:10.13328/j.cnki.jos.005330]

[15] 鲁宁,李峰,王尚广,史闻博,杨放春.一种层次化的基于对等过滤的反匿名联盟构建方法.软件学报,2017.[doi:10.13328/ j.cnki.jos.005517]