ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.10, October 2010, pp.2584−2598
doi: 10.3724/SP.J.1001.2010.03657

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

Co-Monitor:检测前缀劫持的协作监测机制*

刘 欣[1,2+], 朱培栋[1], 彭宇行[1]

[1](国防科学技术大学 计算机学院,湖南 长沙 410073)

[2](中国联通湖南分公司,湖南 长沙 410014)

Co-Monitor: Collaborative Monitoring Mechanism for Detecting Prefix Hijacks

LIU Xin[1,2+], ZHU Pei-Dong[1], PENG Yu-Xing[1]

[1](School of Computer, National University of Defense Technology, Changsha 410073, China)

[2](Hu'nan Branch of China Unicom, Changsha 410014, China)

+ Corresponding author: E-mail: xin.liu@nudt.edu.cn

Liu X, Zhu PD, Peng YX. Co-Monitor: Collaborative monitoring mechanism for detecting prefix hijacks. *Journal of Software*, 2010,21(10):2584−2598. http://www.jos.org.cn/1000-9825/3657.htm

Abstract: In today's Internet, it is very difficult for network operators to discover prefix hijacks in time. Considering the autonomous characteristic of the Internet inter-domain routing system, this paper provides the idea of collaborative monitoring among multiple Autonomous Systems (ASes). This paper also examines the design of a new method, named Co-Monitor that detects prefix hijacks in real-time. In Co-Monitor, every participant AS exchanges self-defined prefix-to-origin mapping information with the others, and they monitor local BGP (border gateway protocol) updates respectively. Once some participant discovers that the origin of information of a BGP route is inconsistent with the learned prefix-to-origin mapping information, it notifies relative participants immediately; thereby, Co-Monitor can help participants detect prefix hijacks quickly and effectively. This paper presents the detailed design of Co-Monitor, evaluates its detecting capabilities, and also discusses several related problems. The experimental results show that Co-Monitor, with only selected 60 participants, is accurate with 0% false negative ratio and 0% false positive ratio.

Key words: BGP; autonomous system; prefix hijacking; source verification; collaborative monitoring

摘 要: 在如今的互联网中,网络管理员要想及时地发现前缀劫持事件非常困难.考虑到互联网域间路由系统中存在的自治特性,提出了在多个自治系统之间协作监测前缀的思想,并由此设计了一个实时检测前缀劫持的新方法——Co-Monitor 机制.在 Co-Monitor 中,每个参与者与其他参与者交换自定义的前缀-源自治系统映射信息,同时,利用所学到的前缀-源自治系统映射信息实时地监测本地 BGP(border gateway protocol)路由更新.一旦某个参与者发现了不一致就立刻通知相关的参与者,从而可帮助参与者及时、有效地发现前缀劫持.给出了 Co-Monitor 机制的详

* Supported by the National Natural Science Foundation of China Grant Nos.60873214, 60433040 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2006AA01Z213, 2006AA01Z332 (国家高技术研究发展计划(863)); the Research Foundation for Ph.D. Candidates of National University of Defense Technology of China under Grant No.B070603 (国防科学技术大学博士研究生创新资助)

Received 2008-08-25; Accepted 2009-05-05

细设计,评估了该机制的检测能力,并讨论了几个相关的问题.实验结果表明,只需精心选择 60 个参与者,就可确保 Co-Monitor 系统检测前缀劫持的漏检率和误检率都为 0%.

关键词: 边界网关协议;自治系统;前缀劫持;宣告方验证;协作监测

中图法分类号: TP393 **文献标识码:** A

互联网由许多相互连接的自治系统(autonomous system,简称 AS)组成,这些自治系统使用边界网关协议(border gateway protocol,简称 BGP)交换各自的网络层可达信息[1].每当一个自治系统向外宣告一条 BGP 路由,它就会把自己的自治系统号附加在该路由的 AS_PATH 属性的尾端,最初宣告该路由的自治系统被称为该路由的源自治系统(origin AS),其自治系统号就位于该路由的 AS_PATH 属性的最右端.BGP 协议规定,BGP 发言人必须无条件地相信其他发言人宣告或传递的路由.因而,一个自治系统可随意地发布其他自治系统拥有的前缀,从而形成前缀劫持攻击.这种路由攻击的关键特征是 BGP 路由中的前缀与源自治系统的对应关系(或称为前缀源信息(prefix origins))不合法[2].

前缀劫持是当前 BGP 路由系统所面临的一个严重的安全威胁,轻则增加路由器负载,重则影响网络连通性.实际观察表明,前缀劫持现象确实经常发生,许多大规模的网络瘫痪事故都与此相关[3−5].值得注意的是,还有许多小型的前缀劫持(只涉及到几个前缀)很少被报道或是难以察觉,它们的安全威胁也不容忽视.举例而言,2005 年 5 月 7 日,Google 中断了近 1 小时的网络服务,Google 事后的解释是错误配置 DNS 服务器所致[6].然而,Wan 等人在研究后发现,网络运营商 Cogent(AS174)在事故期间宣告了 Google(AS15169)的前缀 64.233.161.0/24,他们进而推测 Cogent 对 Google 实施的前缀劫持攻击很可能是此次事故的真正原因[7].另一个小型的前缀劫持事件更具有代表性,它就是曾在互联网上引起轩然大波的 YouTube 服务中断事件.2008 年 2 月 24 日,由于宗教冲突原因,巴基斯坦电信管理局(Pakistan Telecom Authority)命令巴基斯坦的网络运营商阻塞 YouTube(AS36561)的前缀 208.65.153.0/24.但是,巴基斯坦电信(AS17557)泄露了相关路由,并通过香港的电讯盈科(AS3491)在互联网上广为扩散,这次前缀劫持事件造成 YouTube 服务中断了两个多小时[8].

近年来,无论是工业界还是学术界都非常关注 BGP 路由系统中的前缀劫持问题.然而,网络运营商要想及时地发现前缀劫持事件依然十分困难.现有的检测方案要么不够实用,要么能力不足,详见本文第 1 节.我们认为,实用而又有效的前缀劫持检测方案不应忽视 BGP 路由系统的自治特性:在保护各个自治系统的路由信息私密性的前提下,不仅要能帮助网络管理员便捷地获取其他自治系统的相关信息,还要能帮助其对外贡献本地路由域中的相关信息,使它们能够互利互惠地完成各自的前缀劫持检测任务.

由此,本文把每个自治系统对本地路由域的监测能力视为一种资源,提出了协作监测前缀的思想,进而设计了一种实时检测前缀劫持的新方法——Co-Monitor 机制.在 Co-Monitor 中,每个参与其中的自治系统需要与其他参与者交换自定义的前缀-源自治系统映射信息,同时利用所学到的全局的前缀-源自治系统映射信息实时监测本地 BGP 路由更新.一旦某个参与者发现了不一致就立刻通知相关的参与者.通过这种方式,Co-Monitor 机制就可以帮助参与者及时地发现前缀劫持.本文提出的 Co-Monitor 机制具有以下 5 个特点:1) 位于互联网的应用层,除了需要通过哑 iBGP 会话采集 BGP 路由外**,不需要对现有的互联网路由基础设施作任何改变,这便于实际应用与增量部署;2) 利用宣告方验证前缀的思想获得了当前最实用、最准确的前缀劫持判断能力,这样就可保证检测前缀劫持的误检率(false positive ratio)低至 0%;3) 利用协作监测前缀的思想可显著扩展每个参与者的路由监测范围,从而极大地降低检测前缀劫持的漏检率(false negative ratio);4) 每个参与者只需负责监测本地路由域中的 BGP 路由器,这种方式分摊了整个系统的监测开销,因而不存在现有 BGP 监测系统那样严峻的可扩展性问题;5) 各个参与者交换的只是自定义的前缀-源自治系统映射信息,传递的消息中也只含有相关 BGP 路由的前缀源变化信息,都不涉及具体的 BGP 路由,因而不会泄漏任何自治系统的私密信息.现有的前缀劫持检测方案都不同时具备这些优势,这使得它可以帮助网络管理员实时、有效地检测前缀劫持事件.

** 哑 BGP 会话是指只用于接收而不宣告路由更新的 BGP 会话.通过这种会话可在采集 BGP 路由数据时不影响 BGP 路由系统.

本文的工作不能取代完整的互联网域间路由安全方案.我们没有考虑AS_PATH或其他BGP属性的验证问题.特别地,Co-Monitor 机制不能检测由于篡改 AS_PATH 属性而形成的类似于前缀劫持的路由攻击,该机制也不能主动地防范前缀劫持.Co-Monitor 机制仅面向自治系统的网络管理员,帮助他们快速响应前缀劫持事件.

本文第 1 节介绍相关工作.第 2 节概述 Co-Monitor 机制检测前缀劫持的基本原理.第 3 节详述 Co-Monitor 机制的设计方案,其中包括关键前缀、实体认证、消息定义、消息传递、成员管理、拓扑维护与消息抑制等内容.第 4 节评估 Co-Monitor 机制检测前缀劫持的能力.第 5 节讨论相关问题.最后总结全文.

1 相关工作

从实施前缀劫持检测的位置来看,相关工作大致可归为两类:一类是在路由接收方实施的方案[9−17],另一类是在路由宣告方实施的方案[18−23].前一类方案是当前研究的主流,还可进一步细分为防范方案和推断方案.防范方案多是扩展 BGP 协议的安全增强机制,比如 S-BGP[9],soBGP[10]和 psBGP[11]等,尽管它们在理论上可以较好地解决前缀劫持问题,但考虑到协议开销、协议扩展以及设备升级等现实问题,这类方案并不实用;而推断方案可增量部署且不必改动底层的路由设施,但推断结果的漏检率和误检率还有待降低[15,16].第 2 类方案较少,但比较实用,多是当前正在运作的 BGP 监测系统,比如窥镜服务器(looking glasses)[18]、RIPE 的 MyASN 公共服务[20]以及 Renesys 公司的 Gradus 商业服务[21]等.在日常的网络管理中,网络运营商主要使用这类工具诊断 BGP 路由的各种问题.本文的 Co-Monitor 机制属于第 2 类方案,从提升网络运营商前缀监测能力的角度开展研究.

通常,网络运营商会自愿设立窥镜服务器,供他人受限地查询本地 BGP 路由[18].这些数量庞大的窥镜服务器是网络管理员在诊断BGP路由时最常用的工具.但是,网络运营商若要及时地发现前缀劫持,窥镜服务器的作用还很有限:首先,窥镜服务器提供的路由查询服务不是一种标准的服务,并且,由于担心内部路由信息泄露,许多网络运营商对窥镜服务器进行了限制,比如,只允许手工访问等;其次,选择窥镜服务器很有技巧,即使是有经验的网络管理员也需要主动地访问许多窥镜服务器才可能发现前缀劫持,这个过程会耗费大量的时间和精力;最后,正是由于前两个原因,人们往往只是在出现前缀劫持,并引起了网络连通性故障以后,才注意到问题的存在.显然,这种“拉模式”的手工检测方式太过原始,难以帮助网络管理员快速地响应前缀劫持问题.

为减轻网络管理员的负担,业界开发了几个基于“推模式”的 BGP 监测系统,比如 MyASN,Gradus 和 PHAS(prefix hijack alert system)[19]等.现有的这些 BGP 监测系统的体系结构基本相似:它们都由第三方机构设立,都要求与多个自治系统的 BGP 路由器建立多跳步的哑 eBGP 会话,集中地采集、分析 BGP 路由更新,并提供集中式的路由监测服务.这种集中式的体系结构主要存在 3 个缺陷:1) 系统的监测能力严重地受制于系统的处理能力,导致在互联网上存在大片的监测盲区.比如,RouteViews 项目中最大的采集器目前也只与 42 个自治系统的 49 个 BGP 路由器建立了会话[22];欧洲 Netlantis 项目的采集器因接收到太多的 BGP 路由更新而被迫关闭[23].2) 为减少性能开销,系统通常采用分级方式采集 BGP 路由,这会导致采集的 BGP 路由缺乏路由多样性(route diversity).3) 系统要求各自治系统贡献自己的私有 BGP 路由数据,因而难以适应自治的互联网域间路由环境.各个自治系统没有利益动机允许其他机构采集自己的 BGP 路由;相反地,为保护自身利益,它们更可能拒绝建立或仅提供被过滤的 BGP 会话.本文的方案结合了宣告方验证前缀和协作监测前缀的思想,解决了现有 BGP 监测系统面临的这些问题.

2 机制概述

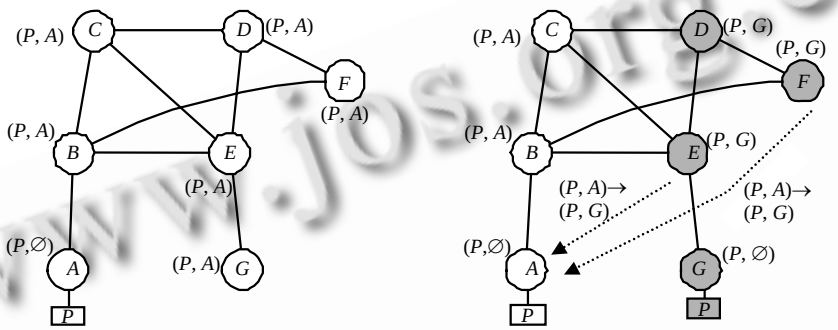
本节概述 Co-Monitor 机制.首先介绍 Co-Monitor 机制检测前缀劫持的基本原理,即该机制中的两个基本思想:一是宣告方验证前缀的思想,它可确保检测前缀劫持的误检率低至 0%;二是协作监测前缀的思想,它可显著降低参与自治系统检测前缀劫持的漏检率;然后给出 Co-Monitor 机制实时检测前缀劫持的基本过程.

2.1 宣告方验证前缀

现有的绝大多数前缀劫持检测方案都要求在路由的接收方进行验证,然而 Atkinson 等人在研究后指出,这

些方案在实际应用中面临着一个难以克服的困难,即不存在完整、准确、真实的信息回答“哪个组织有权利宣告哪个前缀”这个基本问题[24].另一方面,各网络运营商对自己的 IP 地址空间负责,他们通过各种方式关注自己宣告的前缀,共同维护着 BGP 系统的正常运作(本文称其为宣告方验证前缀).我们还注意到,尽管现有 BGP 监测系统在前缀劫持的检测方面存在许多不足,但问题的关键不在于宣告方验证前缀思想本身,而在于现有监测系统的集中式体系结构.更为重要的是,宣告方验证前缀的思想回避了在路由接收方验证路由所面临的困难,它能够提供当前最实用、最准确的前缀劫持判断能力.基于这些认识,我们格外重视宣告方验证前缀的思想.

这里通过一个例子来说明 Co-Monitor 机制如何提供对宣告方验证前缀思想的支持.如图 1 所示,图中的自治系统级网络拓扑由 7 个自治系统(*A*~*G*)组成,其中 AS *A*,*E* 和 *F* 使用 Co-Monitor 机制,AS *A* 是前缀 *P* 的合法宣告者.在前缀 *P* 没有被劫持之前,所有自治系统都学到 AS *A* 宣告的前缀,它们接收到路由的前缀源信息都相同,记作二元组(*P*,*A*);而在 AS *A* 内部,该路由的 AS_PATH 属性为空,其前缀源信息则被记作(*P*,∅),如图 1(a)所示.假若 AS *G* 劫持了 AS *A* 的前缀 *P* 并且 AS *E*,*D* 和 *F* 都接收到该劫持路由,这可能会导致它们观察到的前缀源信息由(*P*,*A*)变为(*P*,*G*),如图 1(b)所示.Co-Monitor 机制要求参与者 *E* 和 *F* 把这样的事件立刻通知给参与者 *A*,从而使得参与者 *A* 能够及时地发现本次前缀劫持事件.由于 Co-Monitor 机制利用了参与者对自己宣告的前缀的辨别能力,因而判断前缀劫持的准确性能够得到保证.



(a) Before prefix *P* was hijacked (b) After prefix *P* was hijacked

(a) 前缀 *P* 被劫持之前 (b) 前缀 *P* 被劫持之后

Fig.1 Idea of source prefix verification in Co-Monitor

图 1 Co-Monitor 机制中宣告方验证前缀的思想

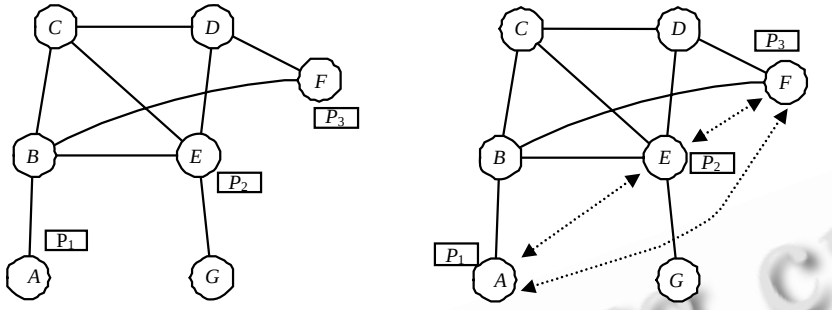
在对宣告方验证前缀思想的支持方面,Co-Monitor 机制与访问窥镜服务器的“拉模式”不同,而与 PHAS 系统和 MyASN 服务的“推模式”相似.但与现有 BGP 监测系统不同的是,Co-Monitor 机制不要求任何自治系统贡献自己的 BGP 路由,参与的自治系统也只是传递 BGP 路由的前缀源的变化信息,没有涉及到具体的 BGP 路由,因而不会泄露任何自治系统的私密路由信息.

2.2 协作监测前缀

直观地,宣告方验证前缀的思想可以保证检测前缀劫持的误检率低至 0%,但问题是,在自治的互联网路由环境中,Co-Monitor 机制没有办法强制一个自治系统为其他自治系统无偿地提供监测服务.比如,在没有得到收益的情况下,图 1 中的参与者 *E* 和 *F* 很可能不愿意为参与者 *A* 监测前缀 *P* 并提供通知消息,从而导致参与者 *A* 检测前缀劫持的漏检率相当高.其实,这也正是当前网络运营商在检测前缀劫持时面临的主要困难.在现实中,某自治系统的网络管理员需广泛了解其他自治系统拥有的相关路由信息,在综合分析以后,才能确定其宣告的前缀是否被劫持,而互联网域间路由系统的自治特性把不同自治系统的网络管理员局限在各自的管理域之内,使他们缺乏了解其他自治系统中 BGP 路由状况的能力,现有的 BGP 监测方案都没有很好地解决这个问题.

为走出这个困境,我们在 Co-Monitor 机制中引入了协作监测前缀的思想.图 2 展示了这个简单却非常有效的思想,它与 P2P 计算所倡导的“我为人人,人人为我”的理念十分类似.这里仍选择 AS *A*,*E* 和 *F* 使用 Co-Monitor 机制,它们分别是前缀 P_1,P_2 和 P_3 的合法宣告者,如图 2(a)所示.由于自治系统对自己拥有的 IP 地址空间负责,

因而 AS *A* 会期望其他所有的自治系统(*B*~*G*)学到关于前缀 P_1 的 BGP 路由都正常,因此正常路由的前缀源信息都应该为二元组(P_1,*A*).同样,对 AS *E* 和 *F* 也是如此,它们分别关心二元组(P_2,*E*)和(P_3,*F*)***.Co-Monitor 机制要求参与者 *E* 和 *F* 根据参与者 *A* 的需要,比如提供的映射关系(P_1,*A*),在各自的路由域中实时监测前缀源信息;若发现不一致,比如,假若前缀源信息变为(P_1,*G*),它们就立刻通知参与者 *A*.同样地,Co-Monitor 机制还要求参与者 *A* 和 *E*(或者,参与者 *A* 和 *F*)也能根据参与者 *F*(或者,参与者 *E*)的需要,实时地监测各自路由域中的 BGP 路由更新.从网络层的角度来看,这些参与者构成了一个层叠型的监测网络,如图 2(b)所示.



(a) Before prefixes were hijacked (b) After prefixes were hijacked

(a) 前缀被劫持之前 (b) 前缀被劫持之后

Fig.2 Idea of collaborative prefix monitoring in Co-Monitor

图 2 Co-Monitor 机制中协作监测前缀的思想

这种协作监测前缀的思想特别适合自治的互联网域间路由环境.Co-Monitor 机制利用该思想激励自治系统加入系统,并为其他参与者提供本地的前缀监测服务.自然地,每个参与者就都可以享受到其他参与者提供的前缀监测服务.在 Co-Monitor 机制的帮助下,所有参与者可相互合作、完成各自的前缀监测任务.

2.3 前缀劫持检测

在介绍了宣告方验证前缀和协作监测前缀的思想以后,下面给出网络运营商利用 Co-Monitor 机制实时检测前缀劫持的基本过程.

整个检测过程由以下 5 个关键步骤组成:1) 每个参与者首先需要与其他参与者交换自定义的前缀-源自治系统映射表(记为 ***L***-table),表中记录了各自的网络管理员所关注的前缀与相应自治系统的对应信息.当然,这个表中的映射信息不一定就是正确的,涉及到的前缀也不一定就是由该自治系统所宣告的,只要是网络管理员所关心的即可.2) 在交换自定义的映射信息以后,每个参与者都动态地维护着一个全局的前缀-源自治系统映射表(或简称为全局映射表,***G***-table).3) 与此同时,每个参与者通过哑 iBGP 会话实时地采集、处理本地路由域中的 BGP 路由更新报文,并检测每条 BGP 路由中的前缀源信息.4) 假若某个参与者发现某 BGP 路由中的前缀源信息与 ***G***-table 中的某些表项不一致,它就会根据这些表项立刻向相关的参与者发出通知消息.5) 每个参与者在接收到来自其他参与者的通知消息以后,还可以根据本地的预警策略向网络管理员发出前缀劫持警报.采用这种协作的前缀监测方式,各个网络运营商就都可以实时地检测前缀劫持事件.

3 机制设计

Co-Monitor 机制涉及 3 类实体:网络管理员、BGP 路由器和监测器(monitor),如图 3 所示.

网络管理员在本文中特指自治系统的管理员,而不是普通网络的管理员,他们维护自身路由域中的 BGP 路由器,并负责建立、配置和使用自己的监测器.BGP 路由器作为网络管理员选择的路由采集点,负责提供本地路由域中的 BGP 路由更新.监测器则是自治系统级别的服务器,负责为网络管理员提供前缀监测服务,其职责包括:通过哑 iBGP 会话采集本地路由域的 BGP 路由、检测路由中的前缀源信息以及在监测器之间传递消息等.

*** 为简化起见,我们没有在图 2 中标出相关的前缀源信息以及前缀源信息的变化.

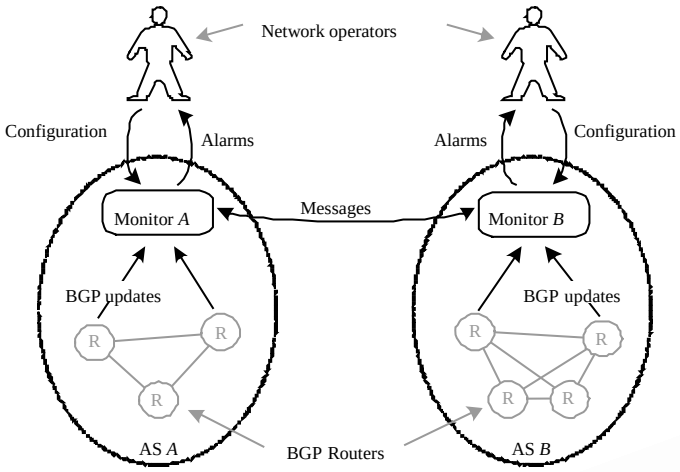


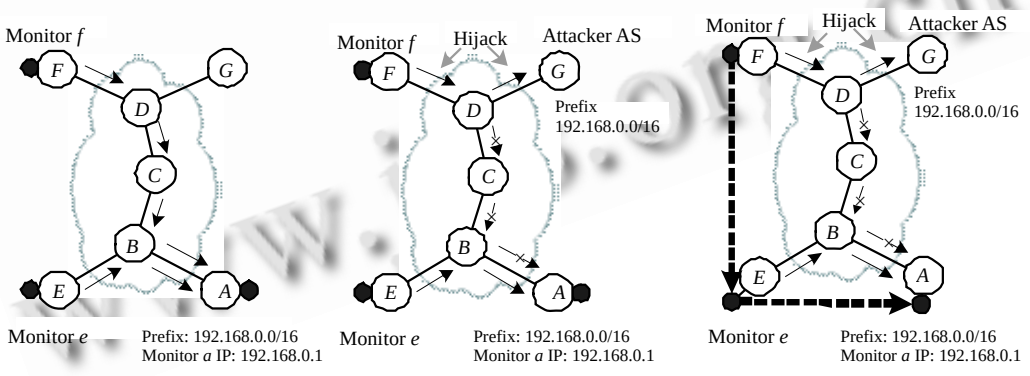
Fig.3 Architecture of the Co-Monitor

图 3 Co-Monitor 机制的体系结构

在实际应用中,Co-Monitor 机制有以下 3 个特点应予以注意:1) 监测器是自治系统级别的服务器,每个自治系统只需设立一个,因而监测器的总数不会很多,不会超过当前自治系统的总数;2) 监测器的存在相对会比较稳定,因为网络管理员不会要求其频繁地加入/退出系统;3) 监测器由网络管理员设立,因而在网络中的位置可受到特别的考虑,比如,可确保拥有足够的网络带宽.尽管如此,Co-Monitor 机制还存在许多问题需要解决,这里主要讨论其中的 3 个重要问题:健壮性、安全性以及通信开销.

- 健壮性

Co-Monitor 机制的健壮性是指该机制本身减缓前缀劫持影响的能力.这非常重要,因为前缀劫持可能中断监测器之间的 IP 层的网络连通性,这会使得监测器不能把通知消息通过 IP 层直接送到正确的目的地.比如,在图 4(a)中,监测器 *e* 和 *f* 在正常的情况下都可以利用 IP 层的网络连通性直接向监测器 *a* 传递通知消息.如果攻击者 AS *G* 劫持了监测器 *a* 的 IP 地址所在的前缀 *P*,并且 AS *F* 选择该劫持路由为最优路由,这时,即使监测器 *f* 检测到了前缀源信息的不一致,它也不能把通知消息送到监测器 *a*;最终,这个通知消息被送往攻击者 AS *G*,如图 4(b)所示.若能利用监测器构成的监测网络来传递通知消息,Co-Monitor 机制就能够获得较好的健壮性.如图 4(c)所示,前缀 *P* 被劫持不会影响到监测器 *f* 与 *e* 以及 *e* 与 *a* 之间的 IP 层的网络连通性,监测器 *f* 可以通过监测器 *e* 间接地把消息送到监测器 *a*.因此,为了获得较好的健壮性,监测器之间需要对通知消息进行路由.



(a) Before monitor *a* was hijacked (b) After monitor *a* was hijacked without flooding (c) After monitor *a* was hijacked with flooding

(a) 监测器 *a* 被劫持之前 (b) 监测器 *a* 被劫持之后,无泛洪 (c) 监测器 *a* 被劫持之后,有泛洪

Fig.4 Influence of prefix hijacking on the robustness of Co-Monitor

图 4 前缀劫持对 Co-Monitor 机制健壮性的影响

• 安全性

Co-Monitor 机制可能会受到来自外部攻击者的恶意攻击.假若图 4 中的攻击者 AS *G* 不仅劫持了 AS *A* 的前缀 *P*,而且还刻意伪造了监测器 *a*,那么 AS *G* 伪造的监测器不仅可以假冒监测器 *a* 接收来自监测器 *f* 的通知消息,而且还可以主动地向其他监测器散发虚假的通知消息,这显然危及到了监测器 *a*,甚至整个系统的安全性.因此, Co-Monitor 机制需要提供实体认证机制防范外部攻击者.另一方面,Co-Monitor 机制还可能会受到来自内部参与者的攻击.假若某个参与者产生了过量的通信开销,或者就是 DoS 攻击,这会影响系统的性能,严重时甚至会使整个系统瘫痪.如何防范来自参与者的攻击,从表面上看,这个问题似乎非常棘手,但实际上,对 Co-Monitor 机制的危害有限,我们在第 5 节会详细讨论该问题.但在这里,我们需要明确,Co-Monitor 机制应该提供消息抑制措施以减缓参与者的 DoS 攻击带来的影响.

• 通信开销

Co-Monitor 机制的健壮性、安全性和通信开销之间相互关联、相互约束,这使得 Co-Monitor 机制的设计变得复杂.考虑到 Co-Monitor 机制工作在互联网的应用层,相对于健壮性和安全性而言,通信开销问题并不太重要.因此,我们的设计原则是:首先确保具有较好的健壮性、安全性,在此基础上,尽可能地减少通信开销.可以看到,在设计 Co-Monitor 机制时,我们会为了获得较好的健壮性和安全性,而牺牲一定的通信开销和计算能力.我们认为,为达到上面的设计目标,这种牺牲不仅必要而且值得.此外,Co-Monitor 机制还需要利用其他措施进一步减少通信开销.比如,Co-Monitor 机制应要求监测器通过已经建立的 TCP 连接尽可能多地传递消息,而又要确保不会传递重复的消息.

下面,我们详细叙述 Co-Monitor 机制中的关键前缀、实体认证、消息定义、消息传递、成员管理、拓扑维护以及消息抑制等内容.

3.1 关键前缀

在当前的互联网中,任何跨自治系统的 IP 网络应用都有可能受到前缀劫持的影响,Co-Monitor 系统当然也不例外.但值得注意的是,并不是所有的前缀劫持都会与某个 IP 网络应用相关.就 Co-Monitor 机制而言,每个监测器的 IP 地址所属的前缀格外重要,如果这些前缀被劫持就极有可能影响监测器之间 IP 层的网络连通性,进而危害 Co-Monitor 系统的健壮性与安全性;而 BGP 路由系统中存在的其他前缀则不然,无论它们是否被劫持都不会对 Co-Monitor 机制产生任何影响.

对于采用 Co-Monitor 机制的某自治系统而言(比如 AS *A*),如果 AS *A* 设立的监测器 *a* 的 IP 地址 *addr* 与 AS *A* 在 BGP 路由系统中宣告的前缀 *P* 满足条件 $addr \in P$,则称前缀 *P* 为 AS *A* 的关键前缀(key prefix).显然,AS *A* 宣告的关键前缀可能存在多个,这可被称为 AS *A* 的关键前缀集(key prefix set).然而,受实际网络运营的影响,比如前缀过滤或聚合等,从路由接收者的角度来看,在不同自治系统上观察到的 AS *A* 的关键前缀集合可能会不一样,而且它们与 AS *A* 宣告的关键前缀集合也可能会不一样.

为便于各个监测器的处理,Co-Monitor 机制要求 AS *A* 宣告的关键前缀集与其在所有自治系统上的出现都一致.为满足这个要求,这里对 AS *A* 在 BGP 路由系统中宣告的关键前缀集有 3 个规定:1) AS *A* 的关键前缀集中含有且仅含有一个关键前缀 *P*;2) AS *A* 的关键前缀 *P* 要能够出现在 BGP 路由系统中的 DFZ(default free zone)区域中;3) 不同自治系统的关键前缀不能相同.由于我们要求自治系统的关键前缀集有且仅有一个关键前缀,为方便起见,在本文随后的叙述中对自治系统的关键前缀集与关键前缀这两个概念不加区别地使用.

3.2 实体认证

我们设计了一个层次式的 PKI 帮助认证参与者,该 PKI 结构如图 5 所示.

参与者需要使用 3 类证书:1) 用于认证各个 RIR(regional Internet registry)(比如 APNIC,ARIN 和 RIPE 等)的区域互联网登记处公钥证书(RIRCert);2) 用于认证自治系统的自治系统公钥证书(ASCert);3) 用于认证监测器的监测器公钥证书(MonitorCert).在该 PKI 中,各 RIR 都可以是认证中心(CAs),因而每个参与者必须信任 RIR 给自己签发的 RIRCert 证书.RIR 需要给自己分配的每一个自治系统号分别签发一个 ASCert 证书,该证书把自

治系统号与一个公钥加以绑定,而该公钥对应的私钥则被相应自治系统的网络管理员所持有.参与者必须给自己设立的监测器签发一个 MonitorCert 证书,该证书把监测器的 IP 地址、相应的关键前缀与一个公钥进行绑定.利用该 PKI 体系,每个参与者都可以获得可信的关键前缀与自治系统的映射信息,这可为 Co-Monitor 提供额外的安全保护.

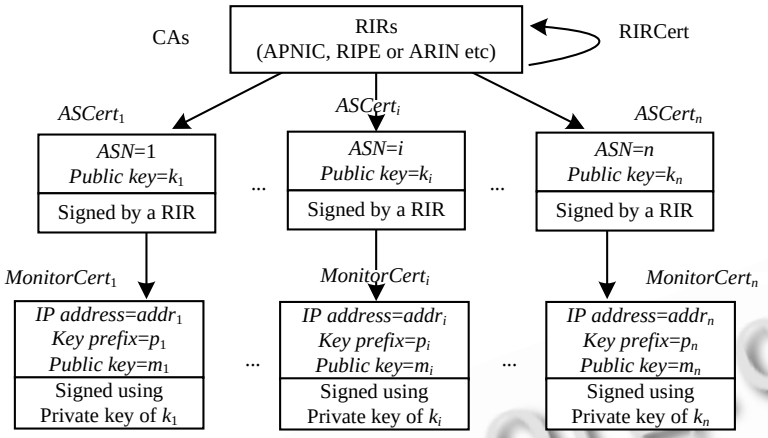


Fig.5 PKI structure in Co-Monitor

图 5 Co-Monitor 机制中的 PKI 结构

由于 ASCert 证书对应的私钥只被其所有者用于签发监测器的 MonitorCert 证书,因此该私钥不必保存在监测器上;而 MonitorCert 证书对应的私钥则必须存储在监测器上,该私钥被监测器用于与其他的监测器建立安全的 TCP 会话,以及对生成的消息进行签名等操作.这种 ASCert 证书与 MonitorCert 证书相分离的设计能够提供额外的安全保护.假若监测器被攻击者侵害,则只会泄露 MonitorCert 证书对应的私钥,而不会对 ASCert 证书及其对应的私钥有任何影响.当然,在出现这种情况后,网络管理员需要撤消并重新签发 MonitorCert 证书.至于对该 PKI 的管理,比如证书的分发和撤消等,可与普通 PKI 的管理相似.由此,我们可以假设每个参与者都能够便捷地获得所有需要的公钥证书.

为确保系统的安全性,每个参与者的监测器都必须执行以下操作:1) 每当一个监测器与另一个监测器建立 TCP 会话时,它们都需要验证对方的身份,如果验证失败则不能建立会话;2) 每当一个监测器生成了一个消息,它都要对该消息进行数字签名;3) 每当一个监测器接收到了一个消息,它都要对该消息的数字签名进行验证,如果失败则丢弃该消息.

3.3 消息定义

在 Co-Monitor 机制中,我们一共定义了 CONNECT_REQUEST,CONNECT_ACK,CONNECT_NAK,DISTRIBUTE,PUSH 和 HELLO 这 6 种消息.前 3 种消息用于维护监测器之间的 TCP 会话连接;后 3 种消息则用于监测器之间交换信息,与前 3 种消息不同,它们会被监测器中转传递.由于 Co-Monitor 机制的语意只与后 3 种消息密切相关,下面仅对它们进行说明:

1) DISTRIBUTE 消息:每当监测器上的 ***L***-table 表被修改,该监测器就要通过 DISTRIBUTE 消息向其他监测器宣告该表的变化情况,该消息的主要作用是分发自定义的前缀-源自治系统映射表内容;

2) PUSH 消息:每当监测器发现本地 BGP 路由中的前缀源信息与 ***G***-table 表中的某些项不一致,该监测器就会立刻向相关的监测器发出 PUSH 消息,该消息的主要作用是通知前缀源信息的不一致事件;

3) HELLO 消息:每当监测器在规定的时间内没有产生任何 DISTRIBUTE 和 PUSH 消息,该监测器就会向其他监测器发送 HELLO 消息,以向其他监测器报告自己依然处于存活状态,该消息的主要作用是辅助成员管理.

为了能够确认消息来源的真实性,消息内容一旦被生成并签名以后,在传播的过程中不应该被改变;而为了控制消息的传播,消息本身又必须携带可被中转监测器修改的内容,比如 TTL(time to live)信息.为满足这种看似

矛盾的需求,我们设计了可变化的消息头(header)和不可变的消息体(body)相结合的消息结构,如图 6 所示.

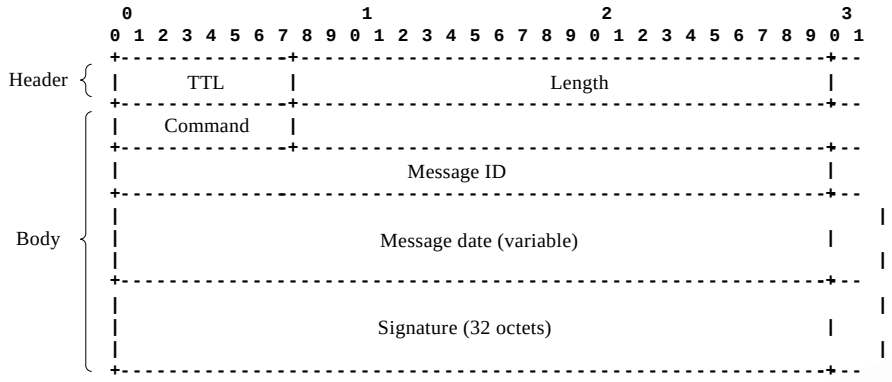


Fig.6 Message structures of DISTRIBUTE, PUSH and HELLO in Co-Monitor

图 6 Co-Monitor 机制中 DISTRIBUTE,PUSH 和 HELLO 的消息结构

消息头含有 TTL 和 Length 两个域:1) TTL 域记录该消息还可被监测器转发的最大跳步数.每当监测器向另一监测器传递消息,都要把该消息的 TTL 值减 1;当 TTL 值为 0 时,该消息不再被转发.2) Length 域记录了该消息的消息体的大小,用于定位消息流中的不同消息.

消息体含有 Command,Message ID,Message Data 和 Signature 这 4 个域:1) Command 域记录了该消息的类型(DISTRIBUTE,PUSH 或 HELLO).2) Message ID 域记录了标识该消息的 ID 号.3) Message Data 域的大小可变,具体的结构与消息类型相关.应当注意,当消息类型是 HELLO 时,该域的大小为零字节.4) Signature 域用于记录该消息的创建者对 Command,Message ID 和 Message Data 这 3 个域中内容的数字签名.

3.4 消息传递

监测器生成的 DISTRIBUTE 和 HELLO 消息需要传递到系统中的所有参与者,而 PUSH 消息只需要传递到系统中的相关参与者.从消息的性质来看,DISTRIBUTE 和 HELLO 消息在 Co-Monitor 机制中是广播消息,而 PUSH 消息则是单播消息.

为了保证具有较好的健壮性、安全性以及较低的通信开销,Co-Monitor 系统自身会维护一个与底层物理网络拓扑相一致的应用层网络.在该网络上,监测器采用标准的泛洪方式(flooding)传递 DISTRIBUTE 和 HELLO 消息.至于 PUSH 消息,监测器采用伪泛洪方式(pseudo-flooding)传递,这种传递方式如图 7 所示.

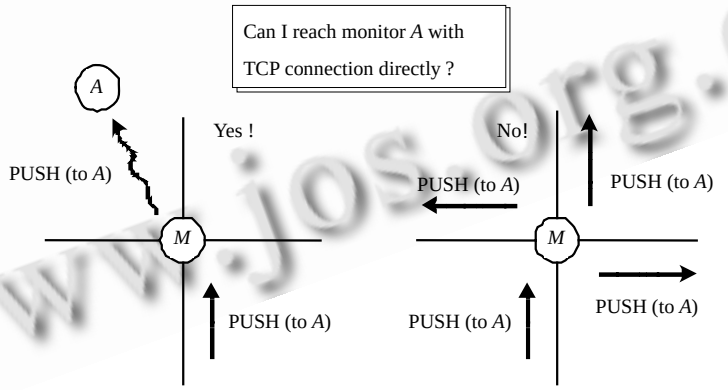


Fig.7 Distributing of PUSH message in Co-Monitor

图 7 Co-Monitor 机制中 PUSH 消息的传递方式

每当一个监测器,比如 *M*,决定转发一个 PUSH 消息的时候,它首先会试图与该消息的目的地,比如 *A*,建立 TCP 会话.如果会话建立成功,监测器 *M* 则会直接把该 PUSH 消息送到目的监测器 *A*,如图 7(a)所示.如果会话没

有建立成功,监测器 M 就会把该 PUSH 消息传递给所有的邻居监测器,请求它们帮助转发,如图 7(b)所示.也就是说,只有在不能通过 IP 层的网络连通性直接送达 PUSH 消息的情况下,监测器才会利用 Co-Monitor 系统维护的应用层网络传递该 PUSH 消息.

3.5 成员管理

实体认证可简化 Co-Monitor 机制中的成员管理工作.当某自治系统决定加入 Co-Monitor 系统时,其网络管理员就可以通过 PKI 的管理体系把 MonitorCert 证书分发给所有的监测器,可以认为这步操作是成员加入(join);而当某自治系统决定退出 Co-Monitor 系统时,其网络管理员就可以把 MonitorCert 证书撤消,从而可以认为这步操作是成员退出(disjoin).然而,某自治系统在分发 MonitorCert 证书以后,可能不会马上设立或者启动监测器;另一方面,某自治系统在关闭监测器后,也可能不会马上撤消 MonitorCert 证书.因此,监测器还需要借助 T_{Message} 定时器和 HELLO 消息跟踪每个监测器的状态,以有效地管理成员.

监测器在启动后,首先要根据获得的所有 MonitorCert 公钥证书构造一张成员列表(**Member**-Table).该成员列表中的每一项是一个五元组〈IPaddr,ASN,KeyPrefix,Status,Time〉对应着一个成员,其中 IPaddr 域为该成员的 IP 地址,ASN 域为该成员的自治系统号,KeyPrefix 域为该成员的关键前缀,Status 域中记录着该成员的当前状态,Time 域中记录着还剩多少时间 T_{Message} 定时器超时(默认为 3 分钟).每个监测器都要记录 Co-Monitor 系统中每个参与者的存活状态,相应的状态转换图如图 8 所示.

Dead 状态代表该成员处于退出系统的状态.在 Active(还是初始状态)或 Neighboring 状态下,只要在规定的时间内,监测器没有接收到源自该成员的任何消息,该成员的状态就转换为 Dead 状态.Active 和 Neighboring 状态都代表该成员处于系统中,如果要停留在这两个状态,监测器则必须在规定的时间内至少接收到源自该参与者的一个消息.特别地,Neighboring 状态指出了监测器与该参与者建立了邻居关系.我们将在下一小节讨论如何选择处于 Active 状态的参与者为邻居.

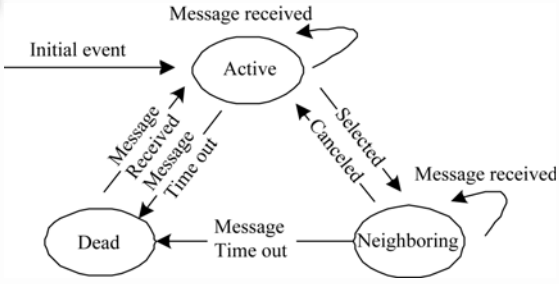


Fig.8 Life state transferring graph of participants in Co-Monitor

图 8 Co-Monitor 机制中参与者的存活状态转换图

3.6 拓扑维护

在 Co-Monitor 机制中,所有监测器共同形成了一个位于互联网应用层的覆盖网络,该网络的拓扑结构不仅对 Co-Monitor 系统自身的健壮性有很大影响,而且也会影响系统给底层 IP 网络带来的通信开销.理想情况下,监测网络的拓扑结构应与互联网的自治系统级拓扑结构相一致.

由于每个监测器都要实时采集本地路由域中的 BGP 路由更新,这有助于 Co-Monitor 机制达到上面的目标.直观地,通过分析每条 BGP 路由中的 AS_PATH 属性信息,监测器就可以构造出互联网的自治系统级拓扑,再结合自身维护的成员列表,就都能选出临近的成员,从而使 Co-Monitor 系统构造的拓扑结构与互联网拓扑相一致.我们注意到,整个 Co-Monitor 系统的拓扑结构在本质上应反映出所有监测器在 IP 层的互连结构,在此基础上,与互联网拓扑结构相一致才有意义.基于这个认识,我们在选择监测器的邻居时仅仅使用含有关键前缀的 BGP 路由,因为非关键前缀的路由与监测器之间的 IP 层的网络连通性无关.此外,考虑到自治系统级拓扑的相对稳定性,监测器并不需要根据 BGP 路由信息完全动态地选择邻居.

我们把监测器的邻居选择任务划分为两个部分:1) 维护关键前缀的 AS_PATH 信息;2) 更新成员列表中成员的 Neighboring 状态.在第 1 部分中,根据实时采集的关键前缀的路由,监测器动态地维护一个容器(aspath_pool),它专门管理 AS_PATH 信息.比如,每当 AS *A* 的监测器 *a* 采集到含有关键前缀的 BGP 路由,就取出 AS_PATH 属性(记作 aspath),然后把其放入该容器.在第 2 部分中,监测器每隔一定时间执行 find_neighbors 算法找出邻居集,然后更新成员列表中相应成员的状态.AS *A* 的监测器 *a* 执行的 find_neighbors 算法如下:

0) 初始构造一个与 *aspath_pool* 同类型的临时容器,记作 *aspath_pool_tmp*,以及一个用于存放 AS 号的临时容器,记作 *as_set*;

1) 依次对 *aspath_pool* 中每个 *aspath* 进行第 2 步处理,其中 $aspath=(AS_1,AS_2,\ldots,AS_n)(n\geqslant 1)$;

2) 根据 *aspath* 中元素顺序生成一个新的 AS_PATH,记作 *aspath_tmp*(注意,*aspath_tmp* 中每个 AS 都存在于监测器 *a* 的成员列表中,并且状态为 Active),然后把 *aspath_tmp* 放入临时容器 *aspath_pool_tmp*,返回第 1 步;

3) 依次对 *aspath_pool_tmp* 中每个 *aspath_tmp* 进行第 4 步处理,其中 $aspath\_tmp=(AS_1,\ldots,AS_m)(1\leqslant m\leqslant n)$;

4) 除 AS_1 不能确定外,把 *aspath_tmp* 中所有非邻居元素 $AS_2,\ldots,AS_m$ 放入临时容器 *as_set*,返回第 3 步;

5) 最后,将监测器 *a* 的成员列表中所有 AS 减去临时容器 *as_set* 中所有 AS,就得到了监测器 *a* 的邻居集.

3.7 消息抑制

监测器有可能无意或恶意地产生大量的转发消息,这会给整个 Co-Monitor 系统带来不利影响.为缓解这种来自合法参与者的问题,Co-Monitor 机制应采取必要措施惩罚那些生成过量转发消息的监测器.我们的基本思路是:在某个时间段内,如果监测器 *a* 接收到源自监测器 *b* 的过量消息,监测器 *a* 就给予监测器 *b* 一段时间的惩罚,在这个时间内,监测器 *a* 不会再转发源自监测器 *b* 的任何消息.

为实现这个消息抑制措施,我们需要在成员列表中加入 3 个附加域,它们分别是时间记数器 *T*、消息记数器 *C* 以及惩罚时间记数器 *R*.根据实际情况的不同,网络管理员可以给不同监测器的这 3 个记数器赋予不同的初值.比如,监测器 *a* 的网络管理员可以给监测器 *b* 赋予的记数器初值分别为 *T*=600(s),*C*=1000 个,*R*=300(s).至于监测器 *b* 在正常状态和惩罚状态之间的转换与动作细节,设计起来并不困难,这里不再赘述.

4 实验与评估

通常,人们通过误检率和漏检率这两个关键指标评估监测系统的检测能力.在前面的讨论中,我们曾指出,由于 Co-Monitor 系统自身的特点使得其检测前缀劫持的误检率为 0%;然而,对 Co-Monitor 系统检测前缀劫持漏检率的讨论却要复杂得多.本节首先评估影响漏检率的关键因素——监测范围;然后再评估漏检率.

4.1 监测范围

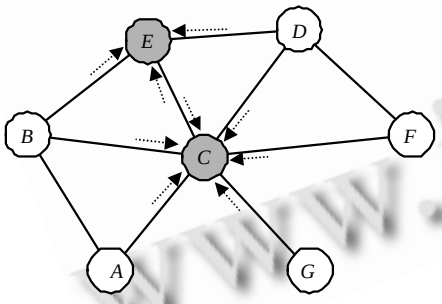


Fig.9 Monitoring boundary of single AS

图 9 单个自治系统的监测边界

自治系统的网络管理员在互联网中所能达到的路由监测范围是能否检测出前缀劫持的关键因素.在自治的互联网中,一个自治系统的网络管理员通常只能监测本地路由域,如图 9 所示.这个特点对网络管理员监测前缀劫持极为不利,因为他们还需要广泛了解其他自治系统中的 BGP 路由状态才能发现自己的前缀是否被劫持.好的监测方案要能帮助单个自治系统扩大在互联网上的监测范围.

可令 INT 代表互联网中自治系统的全集.对于互联网中的任意一个自治系统而言,比如 AS *A*,令其邻居自治系统集为 S_A.在实际的网络运营中,自治系统在本地路由域中可监测到来自邻居宣告的 BGP 路由.显然,一个自治系统的邻居越多,它的监测范围也就越广.比如,图 9 中 AS *C* 的监测范围就要比 AS *A* 的监测范围大.因此,在通常情况下,可用 AS *A* 的邻居数与其自治系统邻居数的比值来定义 AS *A* 的监测范围,记作 V_A,即

$$V_A=\frac{|S_A|}{\sum_{\alpha\in INT}|S_\alpha|},\ V_A\in[0,1] \tag{1}$$

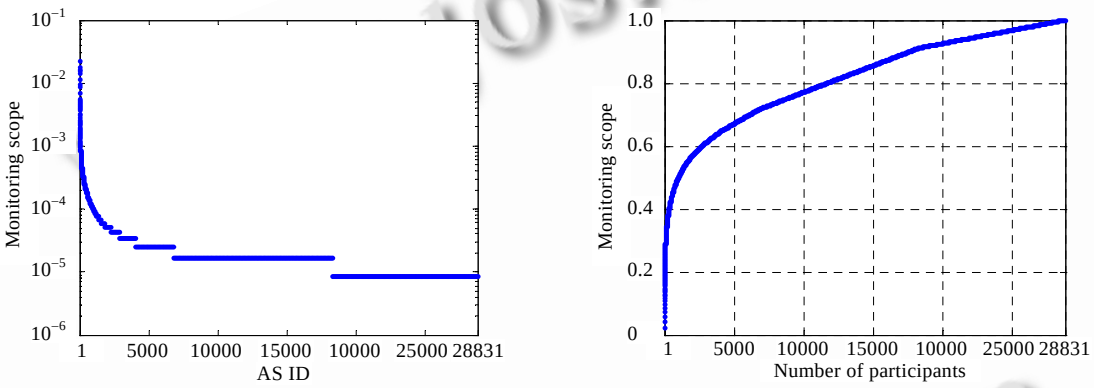
在公式(1)中,当 V_A=0 时,表示 AS *A* 不能监测互联网的任何部分,也就是说,AS *A* 没有与互联网相连;而由于 AS *A* 不可能监测到整个互联网,因此,V_A 不能为 1.

可令加入 Co-Monitor 系统的自治系统集合为 *COM*.当 AS *A* 不在集合 *COM* 中时,其监测范围满足公式(1);但是,当 AS *A* 在集合 *COM* 中时,由本文 Co-Monitor 机制的特点,*COM* 集合中的所有自治系统都会分享各自的监测范围.因此,AS *A* 的监测范围会扩展到 *COM* 集合中的所有自治系统,即可得到如下公式:

$$V_A=\frac{\sum_{\beta\in COM}|S_\beta|}{\sum_{\alpha\in INT}|S_\alpha|},\ V_A\in[0,1] \tag{2}$$

在公式(2)中,当 V_A=0 时,表示 *COM* 集合中仅有 AS *A* 并且该 AS 没有与互联网相连;而与公式(1)不同的是,V_A 可以为 1,因为当所有自治系统都属于 *COM* 集合时,AS *A* 就可以监测整个互联网.

利用式(1)和式(2),我们使用 RouteViews 提供的 BGP 路由数据[22],具体选用路由表快照的时间点为 2007 年 6 月 20 日 10 时,对自治系统加入与不加入 Co-Monitor 系统所具有的监测范围进行评估.为突出效果,我们把各自治系统按邻居数由大到小排序,并按 1~28 831 的顺序依次赋予 ID 值.图 10(a)展示了互联网上的各自治系统在不加入 Co-Monitor 系统情况下,各自所具有的监测范围值;而图 10(b)则展示了随着加入 Co-Monitor 系统的自治系统数目的增长(按 ID 值从小到大的顺序依次加入),各参与者所具有监测范围值的变化情况.



(a) Monitoring scope without Co-Monitor (b) Monitoring scope with Co-Monitor

(a) 不加入 Co-Monitor 时的监测范围 (b) 加入 Co-Monitor 时的监测范围

Fig.10 Comparison of monitoring scope of AS between Co-Monitor and non-Co-Mointor

图 10 AS 加入 Co-Monitor 与不加入 Co-Mointor 的监测范围比较

对比图 10(a)和图 10(b)可以看到,当不加入 Co-Monitor 系统时,邻居数最多的自治系统(AS 701,其 ID=1)的监测范围为 2.2%,而绝大多数自治系统的监测范围非常小,几乎为 0;当加入 Co-Monitor 系统时,单个自治系统的监测范围可随整个系统中参与者的增加而显著扩大.特别地,只要最大的 10 个自治系统采用本文方案,系统中的每个自治系统就至少可监测到当前互联网的 13%以上;为了监测到互联网范围的 50%,也只需要 900 多个自治系统加入 Co-Monitor 系统即可,而这个数目还占不到当前自治系统总数的 4%.

4.2 漏检率

直观地,越多的自治系统加入 Co-Monitor 系统,系统中每个参与者的监测范围越大,检测前缀劫持的漏检率就会越小;另一方面,Co-Monitor 系统是否能够检测到互联网中的前缀劫持事件,不仅与系统中参与者集合的增长方式有关(影响监测范围),还取决于前缀劫持发生的位置以及传播的途径等多方面因素.本节主要评估 Co-Monitor 系统的漏检率与参与者集合增长方式之间的关系.

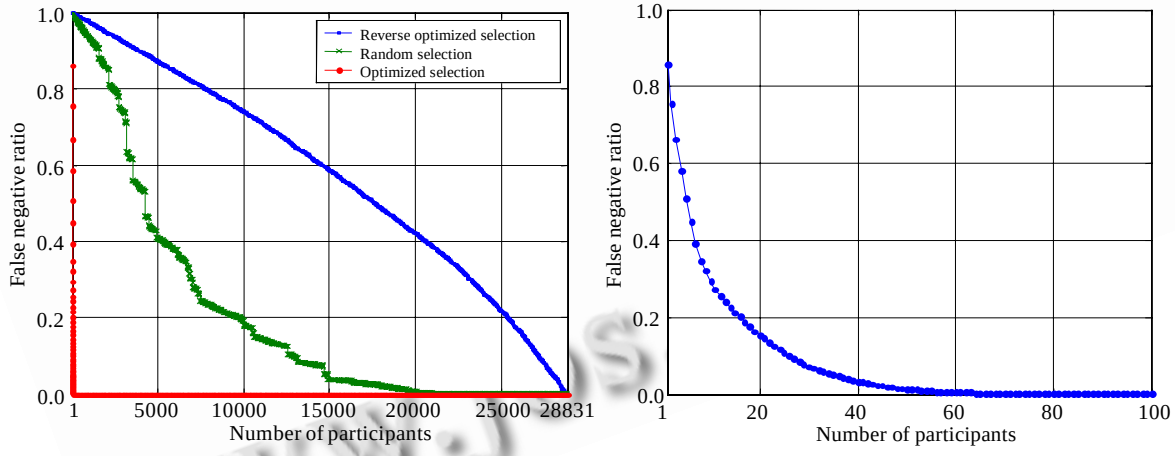
首先,为便于评估 Co-Monitor 系统的漏检率,我们需要对如下两个问题做出合理的假设:1) Co-Monitor 系统

如何增长?即参与自治系统(监测点)的选取问题;2) 前缀被劫持的路由如何在互联网上传播.对于前一个问题,我们假设在 BGP 路由系统中,宣告越多前缀的自治系统就越会期望加入 Co-Monitor 系统,因为它们期望 Co-Monitor 系统为它们宣告的前缀提供实时的前缀监测服务.基于这样的考虑,我们在选取监测点时,最优采取策略是:首先依次选取宣告前缀数目多的自治系统;在前缀数目相等的情况下,再依次选取邻居数目多的自治系统;而如果两个自治系统宣告的前缀数和邻居数都一样,则选取自治系统号小的那个自治系统.文献[25]指出,在不考虑路由聚合和路由安全过滤的情况下,劫持路由和正常路由都是按相同的方式传播.因此对于第 2 个问题,我们假设任何自治系统都可能劫持别人的某个前缀且概率均等,并且,任何合法的自治系统路径都有可能是劫持路由的传播途径且概率均等.显然,基于这样宽泛的假设,我们评估的是检测前缀劫持的漏检率的上界.

可令 BGP 路由系统中所有的有效自治系统路径所构成的集合为 *PATHS*(这里所说的有效路径是指满足自治系统商业互连关系的无谷底(no valley)路径[26]),而令 *PATHS* 集合中所有涉及 Co-Monitor 系统中参与者的路径集合为 $PATHS_{COM}$(显然 $PATHS_{COM} \subseteq PATHS$).由 Co-Monitor 系统检测前缀劫持的特点可知,在各参与者确保自定义的前缀-源自治系统映射表完整、正确的前提下,在 $PATHS_{COM}$ 中任何一条路径上传播的劫持路由都可能被 Co-Monitor 系统发现,由此可得如下漏检率计算公式:

$$1-\frac{|PATH_{COM}|}{|PATHS|} \tag{3}$$

图 11 展示了 Co-Monitor 系统检测前缀劫持的漏检率随参与者数目增长而发生变化的情况.实验数据来自 RouteViews 公布的 2008 年 06 月 30 日 08 时 BGP 路由表,其中共有 28 821 个自治系统.这里考虑了 3 种选取参与者的策略:1) 前面所说的最优采取策略(optimized selection);2) 最糟糕的选取策略,也就是最优选取策略的逆过程(reverse optimized selection);3) 随机选取策略(random selection).图 11(a)给出了这 3 种选取策略下的漏检率变化情况.可以看到,不同的选取策略对 Co-Monitor 系统检测前缀劫持的漏检率影响极大,特别是最优选取策略,几乎可以确保漏检率为 0%.为进一步了解在最优采取策略下漏检率的变化情况,图 11(b)只给出了参与者数从 1 增长到 100 时漏检率的变化情况.由图 11(b)可以看到,当 Co-Monitor 系统中有 20 个参与者时,漏检率仅为 15.16%;当有 40 个参与者时,漏检率就可低至 3.32%;而只要有 60 个参与者时,漏检率几乎为 0%.



(a) False negative ratio under different selection policies
(a) 在不同选择策略下的漏检率变化情况

(b) False negative ratio under optimized selection
(b) 在最优采取策略下的漏检率变化情况

Fig.11 Comparison of false negative ratio under different selection policies
图 11 不同选择策略下的漏检率变化情况比较

5 相关讨论

本节讨论几个与 Co-Monitor 机制相关的问题.

(1) PKI 结构的实用性问题.S-BGP 协议中需要两个 PKI 结构分别用于认证自治系统号和网络前缀的所有

权.文献[11]指出,用于认证自治系统号的 PKI 规模不大,实际部署可行.但是,用于认证网络前缀的 PKI 结构过于庞大,这是 S-BGP 协议难以被实际应用的重要原因之一.本文不存在认证网络前缀的问题,而且与 S-BGP 协议中认证自治系统号的 PKI 结构相比,存在以下 4 个优势:1) 以自治系统为签发对象而不是网络运营商,本文的 PKI 结构的稳定性更好;2) ASCert 证书的签发方式与当前自治系统号码资源的管理模式相一致,并且本文的 PKI 结构忽略了 ICANN,这样的 PKI 结构更为简单和实用;3) 整个 PKI 结构涉及的证书数量不多,包括 5 个 RIR 的 RIRCert 证书、所有参与自治系统的 ASCert 证书以及相应的 MonitorCert 证书(总数为 $5+2\times N$,其中 N 为参与者数);4) 由于 Co-Monitor 机制位于互联网的应用层,利用 PKI 对其进行安全保护不存在部署与应用上的困难.

(2) 参与者的恶意攻击问题.Co-Monitor 机制通过实体认证可以有效防范来自系统之外的恶意攻击,但合法的参与者也有可能执行危害系统的行为.比如,参与者可能恶意丢弃消息、发送虚假的消息或是 DoS 攻击等.我们要阻止来自参与者的这些恶意攻击非常困难,因为它们都拥有合法的身份.但是,这些恶意行为对 Co-Monitor 机制的影响不大,原因如下:① 前缀被劫持的路由通常会被扩散到多个自治系统,这被多个参与者观察到的概率极大,少数恶意参与者丢弃消息或发送虚假的消息并不会影响到其他多数参与者发送真实的消息;② 我们利用消息抑制措施可以缓解 DoS 攻击的影响,并且系统中的任何消息都经过签名,因而可知 DoS 攻击的确切来源.

(3) 搭便车(free riding)问题.一般的 P2P 应用中都存在搭便车的问题,即某些对等体只愿享受别人提供的服务,自己却不愿为别人提供服务.在 Co-Monitor 机制的实际应用中也可能出现类似的现象,即可能会有某些参与者只愿享受来自其他参与者提供的前缀监测服务,自己却不愿为其他参与者提供前缀监测服务.具体做法是,这些自治系统的网络管理员不在监测器与本地路由域中的 BGP 路由器之间配置 BGP 会话.但我们认为,搭便车问题在 Co-Monitor 机制中并不是一个问题:首先,Co-Monitor 机制不会泄露参与者的私密路由信息,这在前面已经说明过;其次,参与者为其他参与者提供前缀监测服务,不仅对他人有益,而且对其自己也是有益的,因为这可以让他人帮助检查自己接收的路由;最后,如果参与者还是不愿意提供前缀监测服务,这也没有问题,只要其加入了 Co-Monitor 系统,它就会为成为覆盖网络的一部分,从而可增强系统的健壮性.

6 结论及工作展望

本文结合协作监测前缀和宣告方验证前缀的思想解决了现有 BGP 监测系统所面临的问题.本文提出的 Co-Monitor 机制适应当前自治的互联网环境,可逐步地实际部署,不需要对 BGP 协议进行任何安全扩展,并且能够保护参与者的私密路由信息,这些特性使其可以有效地帮助网络管理员实时地检测前缀劫持事件.

我们将来的工作主要集中在 Co-Monitor 机制的实际应用与部署等方面.另外,考虑利用 Co-Monitor 平台支持更丰富的自治系统协作工作也是我们着重要关注的问题.

致谢 感谢审稿人对论文初稿提出的宝贵意见.

References:

[1] Rekhter Y, Li T, Hares S. A border gateway protocol 4 (BGP-4). RFC 4271, IETF, 2006.
[2] Nordström O, Dovrolis C. Beware of BGP attacks. ACM SIGCOMM Computer Communications Review, 2004,34(2):1−8.
[3] Bono VJ. 7007 Explanation and apology. 1997. http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html
[4] Popescu AC, Premore BJ, Underwood T. Anatomy of a leak: AS9121. 2005. http://www.nanog.org/mtg-0505/underwood.html
[5] Farrar J. C&W routing instability. 2001. http://www.merit.edu/mail.archives/nanog/2001-04/msg00209.htm
[6] Ferguson P. Google DNS problems. 2005. http://www.merit.edu/mail.archives/nanog/2005-05/msg00238.html
[7] Wan T, Oorschot PC. Analysis of BGP prefix origins during Google's May 2005 outage. In: Proc. of the 20th IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS 2006). Washington: IEEE Computer Society Press, 2006. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1639679
[8] Pakistan hijacks YouTube. 2008. http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml
[9] Kent S, Lynn C, Seo K. Secure border gateway protocol (S-BGP). IEEE Journal on Selected Areas in Communications Special Issue on Network Security, 2000,18(4):582−592.

[10] White R. Securing BGP through secure origin BGP. Internet Protocol Journal, 2003,6(3):15−22.
[11] Kranakis E, Wan T, Oorschot PC. On interdomain routing security and pretty secure BGP (psBGP). ACM Trans. on Information and System Security, 2007,10(3):1−41.
[12] Xu K, Xiong YQ, Wu JP. Security extension of border gateway protocol BGP-4. Acta Electronica Sinica, 2002,30(2):271−273 (in Chinese with English abstract).
[13] Liu X, Zhu PD. A rules-based approach to anomaly detection in inter-domain routing system. Journal of National University of Defense Technology, 2006,28(3):71−76 (in Chinese with English abstract).
[14] Hu XJ, Zhu PD, Gong ZH. SE-BGP: An approach for BGP security. Journal of Software, 2008,19(1):167−176 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/167.htm [doi: 10.3724/SP.J.1001.2008.00167]
[15] Karlin J, Forrest S, Rexford J. Pretty good BGP: Improving BGP by cautiously adopting routes. In: David L, ed. Proc. of the IEEE Int'l Conf. on Network Protocols. Washington: IEEE Computer Society Press, 2006. 283−292.
[16] Zheng CX, Ji LS, Pei D, Wang J, Francis P. A light-weight distributed scheme for detecting IP prefix hijacks in real-time. In: Proc. of the ACM SIGCOMM 2007. 2007. http://portal.acm.org/citation.cfm?id=1282412
[17] Liu X, Zhu PD, Peng YX. Internet registry mechanism for preventing prefix hijacks. Journal of Software, 2009,20(3):620−629 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3221.htm [doi: 10.3724/SP.J.1001.2009.03221]
[18] Looking Glasses. 2008. http://www.traceroute.org
[19] Lad M, Massey D, Pei D, Wu YG, Zhang BC, Zhang LX. PHAS: A prefix hijack alert system. In: Proc. of the 15th USENIX Security Symp. Berkeley: USENIX Association, 2006. 153−166. http://www.usenix.org/events/sec06/tech/full_papers/lad/lad_html/
[20] Ripe myasn system. 2008. http://www.ris.ripe.net/myasn.html
[21] Gradus tool. 2008. http://gradus.renesys.com
[22] Meyer D. Route views project. 2008. http://www.routeviews.org
[23] Gloor P. Netlantis project. 2008. http://www.netlantis.org
[24] Atkinson R, Floyd S. IAB concerns & recommendations regarding Internet research & evolution. RFC 3869, IETF, 2004.
[25] Lad M, Oliveira R, Zhang BC, Zhang LX. Understanding resiliency of Internet topology against prefix hijack attacks. In: Proc. of the 37th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks. Washington: IEEE Computer Society, 2007. 368−377. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4272988
[26] Gao LX. On inferring autonomous system relationships in the Internet. IEEE/ACM Trans. on Networking, 2001,9(6): 733−745. [doi: 10.1109/90.974527]

附中文参考文献:

[12] 徐恪,熊勇强,吴建平.边界网关协议 BGP-4 的安全扩展.电子学报,2002,30(2):271−273.
[13] 刘欣,朱培栋.基于规则的域间路由系统异常检测.国防科学技术大学学报,2006,28(3):71−76.
[14] 胡湘江,朱培栋,龚正虎.SE-BGP:一种 BGP 安全机制.软件学报,2008,19(1):167−176. http://www.jos.org.cn/1000-9825/19/167.htm [doi: 10.3724/SP.J.1001.2008.00167]
[17] 刘欣,朱培栋,彭宇行.防范前缀劫持的互联网注册机制.软件学报,2009,20(3):620−629. http://www.jos.org.cn/1000-9825/3221.htm [doi: 10.3724/SP.J.1001.2009.03221]

刘欣(1978－),男,湖南常德人,博士,CCF 高级会员,主要研究领域为互联网域间路由.

朱培栋(1971－),男,博士,副教授,CCF 会员,主要研究领域为路由技术,移动网络,网络安全.

彭宇行(1963－),男,博士,教授,博士生导师,主要研究领域为并行与分布处理技术,计算机网络技术.