ISSN 1000-0054 清华大学学报（自然科学版） 2018 年 第 58 卷 第 8 期 1/12
CN 11-2223/N J Tsinghua Univ (Sci & Technol), 2018, Vol. 58, No. 8 693-697

范式路由器：规范路由器数据层的动态行为

徐 磊，徐 恪

（清华大学 计算机科学与技术系，信息科学与技术国家实验室，北京 100084）

摘 要：随着模块化可编程路由器越来越普遍，路由器面临的安全问题也越来越严峻。该文提出范式路由器，通过对模块化的数据层进行编码和预组合，达到对路由器数据层的动态监控和规范。该文对每个数据层行为标记一个行为标识(action identifier，AID)，同时将合法 AID 预先存入范式表(regulated action table，RAT)。在路由器运行时，所有动态行为都被 RAT 校验，保证行为可信。该文用 Click 路由器和数据层开发包(data plane development kit，DPDK)路由器分别部署了范式路由器。实验结果表明：范式路由器仅占用了 2 MB 的空间和 10% 以下的带宽性能，同时捕获了所有数据层的恶意行为。

关键词：路由器安全；范式路由器；路由器行为

中图分类号：TP393.0 **文献标志码**：A

文章编号：1000-0054(2018)08-693-05

DOI：10.16511/j.cnki.qhdxxb.2018.21.019

Pattern router to regulate dynamic actions in the router dataplane

XU Lei, XU Ke

(Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, China)

Abstract: Router security has become more important with the increasing number of programmable routers. This paper presents a pattern router that codes the modularized dataplane and pre-combines the result to monitor and regulate the dynamic actions in the dataplane. This method uses an action identifier (AID) for each action in the dataplane and puts the normal AID into a regulated action table (RAT) before running the router. When the router is working, all the dynamic actions are verified by the RAT to secure the honesty of each action. The pattern router was implemented in a Click router and in a data plane development kit (DPDK) router with tests showing that the pattern router occupies only 2 MB and uses less than 10% of the bandwidth to capture all the abnormal actions in the dataplane.

Key words: router security; pattern router; router action

过去十年，路由器的容量和性能有了长足的发展[1-2]，其分析模型也有了很大进步[3-4]。可编程路由器正不断涌现[5]，并更加容易部署在新型网络拓扑中，但同时也伴随着安全隐患增加的风险。

最近几年，运行软件部署分组处理功能的路由器越来越容易受到攻击[6-12]。例如，思科路由器被报道发现了设备后门[13-14]，引起了政府部门和运营商很大关注。即使在路由器中所有模块和模块的调用被确认是正常的，模块也会被用来完成不期望的行为。例如，黑客能够获取管理权来操纵路由器行为或通过更新模块来改变路由器行为[15]。

以往的工作专注于代码安全和加密技术[16-17,20]，用来促进路由器行为可信性。Dobrescu 等提出了静态验证工具来检查软件数据层的安全隐患[16]。本文的范式路由器也是检查工具，但检查的是动态运行的数据层行为。Kim 等使用源认证和路径验证来预防路由器的异常行为[17]。密钥算法需要部署代价和验证算法，对于路由器和端主机都要修改，成本较高，本文的范式路由器没有加密机制。

本文设计了在高速运行的路由器动态数据层之上，实时验证路由器行为的验证框架。首先，设计了验证局部构件取代了验证整体行为的思想，具有很高的性能优势。其次，使用集合和图论相关方法优化了验证思想，使得验证更加有效。然后，设计了位图的查找结构，占用了很少的存储容量。最后，设计了初始化机制，将白名单行为集合转化为范式表。

1 构件和行为的验证思想

本文调研了路由器数据层的构件和行为，并且在此观察基础上设计了构件标识（component

收稿日期：2018-03-10

作者简介：徐磊(1983—)，男，博士研究生。

通信作者：徐恪，教授，E-mail：xuke@tsinghua.edu.cn

identifier，CID)和行为标识(action identifier，AID)来描述构件和行为。

1.1 路由器数据层的特点

本文专注于软件路由器平台的安全问题。软件路由器平台由丰富的构件构成，本文的构件是数据层的功能模块，具有以下特征：首先，构件是分组相关的处理模块，它直接对分组进行操作；其次，它是在动态运行数据层的实时处理单元。因此一个行为被看做是构件的管道线。

通过调查 Click 路由器数据层的构件，本文将2类处理单元视为构件：一类是其成员函数对分组进行操作的元素，另一类是 Packet 类里的所有成员函数。通过运行 TCP(transmission control protocol)和 UDP(user datagram protocol)2个协议，从构件调用关系、构件特点、路由器行为特点3个方面观察了路由器行为，得到如下结论：1)路由器行为的最后一个构件必定是 SendPacket 或 KillPacket，2)路由器数据层在运行时并不会调用很多构件，3)规范的路由器行为的集合仅仅是所有构件的组合的一小部分，4)一个行为内的构件数很少。

1.2 CTAG 和 WAID 设计

本文用 CID 来标识构件，用 AID 来标识行为。因为二进制数占用空间少并且可以快速计算查找，基于节 1.1 的调研，本文使用 8 位二进制数描述 CID、构件权重(component weight，CW)、边权重(edge weight，EW)和行为权重(action weight，AW)。图 1 为 CTAG(component tag)和带有权重的 AID(WAID)的格式。其中：AID 是一个行为内调用的所有构件 CID 的有序序列，AW 是一个行为内调用的所有构件的 CW 和 EW 之和。

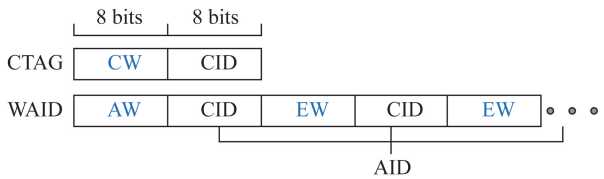


图 1 CTAG 和 WAID 格式

1.3 验证的高效性

AID 暗示了 CID 之间的关系。为了验证当前 AID 是否是正常行为，本文提出验证两两相邻 CID 的关系。这样验证的代价比验证整个行为 AID 的要低。正常的 CID-CID 关系在 RAT(ragulated action table)中作为验证标准。两两验证有效的前提是所有 AID 满足 2 个条件：AID 内部没有有向环，AID 之间没有交互。

1.4 验证的有效性

1)构件和边权重。

当正常 AID 有交互时，两两验证会失效。例如，2 个正常 AID 分别是 4-ABCE$_1$ 和 4-ADCE$_2$，字母代表构件的 CID，数字为 AW，每个构件 CW 为 1，EW 为 0。因此 AB、BC、CE$_1$ 和 AD、DC、CE$_2$ 是正常调用关系。但是当恶意行为 ABCE$_2$ 和 ADCE$_1$ 执行时，RAT 会通过验证，认为是正常行为。

上述 2 个行为有相同 AW 并且他们还有交集(头尾除外)，就叫冲突行为。为解决冲突行为，本文对非共享节点 CID 修改 CW，并将 RAT 的条目变成 AW-CID-CID。在上面的例子中，B、D、E$_1$ 和 E$_2$ 是非共享 CID。通过对非共享的 D 的 CW 加 1，变为 2，则 4-ABCE$_1$ 和 5-ADCE$_2$ 的 AW 分别变成 4 和 5，2 个行为不再冲突。

当冲突的 AID 完全互相共享构件 CID 时，仅靠修改非共享节点的 CW 无法消除冲突行为。本文通过修改非共享边的 EW，可以消除冲突行为。原理和使用 CW 的原理相同。分配权重的方法见节 2.4。

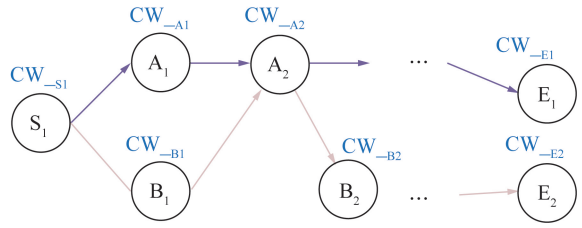


图 2 权重的一般性证明

引理 1 对于带权有向图 G，若 G 中所有的 WAID 互相不冲突，而且没有环，则带权验证两两关系，就可以确定一个行为的合法性。

证明：如图 2 所示带权关系拓扑 G，G 中的带权行为 WAID 满足不冲突且无环。G 中所有带权有向边构成了集合 RAT。使用 RAT 重新遍历组合成新的所有可能行为，形成拓扑 G'。根据构建关系，G' 包含 G。现在用反证法证明 G 包含 G'。不妨设 α_w 来自 G' 且从 S_1 开始并在 E_2 结束，且不属于 G，将 α_w 分为 2 段 S_1-A_2 和 A_2-E_2，S_1-A_2 属于图中最上面的 S_1-E_1，A_2-E_2 属于 S_1-E2，由于 G 无冲突，所以 S_1-E_1 和 S_1-E_2 有不同的权，即 α_w 有 2 个不同的权，矛盾。因此 G 包含 G'。最终 $G=G'$。

2)正常 AID 中的环。

当一个行为内部有有向环时，恶意行为仍然可

以逃过检查。例如，正常行为 6-ABACAE，恶意行为 6-ACABAE 会逃过检查，因为 2 个行为边和节点完全共享，AW 也相等。

为了解决这个问题，本文提出去除环的方法。首先当拼接 CID 组成 AID 时，先检查当前 CID 是否已经存在于要拼接的 AID 中。一旦发现同样的 CID，就增加当前的 CID 序号。然后将当前 CID 加入 AID 尾部。最终计算行为的 AW。这样同一个构件就拥有不同的 CID，也就去除了环。上述例子中，去除环以后，最终的行为是 6-ADBECG。

可以证明上述去环方法不会产生歧义或重复。上述方法定义为映射 $f(a)=A$，$a\in$ AID，$A\in$ RAID。f 定义为从左到右遍历 a，当 a 存在重复节点 C1，如…C1…C1…C1…时，f 会对重复节点号增加 k 值，k 是重复次数。于是 A 是…C10…C11…C12…，其中 C1j=C1+j，$0\leqslant j\leqslant T_1$，$T_1$ 是 C1 在所有行为中的最大重复次数。C10，…，C1T_1 构成集合 γ_1。集合 γ_i 中，$0\leqslant i\leqslant N$，N 是所有原始行为不同节点的数量，$\gamma_0=\varnothing$。可以证明当所有集合 γ_i 互不相交时，f 是可逆映射。证明方法用可逆映射具有单射和满射性质，对于单射：AID 不同的编码不同的重复次数会按照 f 会得到不同的 RAID 的编码，因为 γ_i 互不相交。对于满射，由于因为 γ_i 互不相交，对应的原始编码也不相等，故满射成立。证毕。

2　范式路由器设计

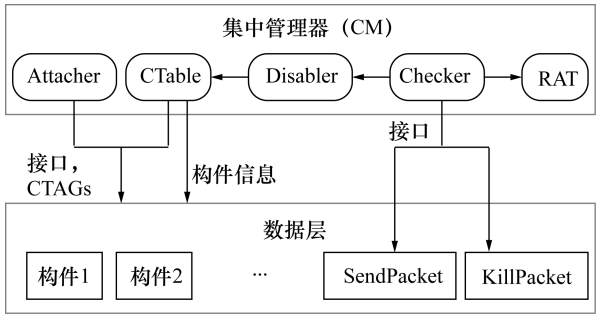


图 3　范式路由器结构

2.1　范式路由器架构

如图 3 所示，本文将验证方法部署在范式路由器，核心结构是集中管理器（centralized manager，CM），它由 5 部分构成：Attacher 模块是在临时分组头部附加 CTAGs 到 WAID 尾部；Ctable 存储了所有的 CTAGs；RAT 是一个位图结构，存储了所有正常的 WAID；Checker 是检查 WAID 正确性的函数；Disabler 则用来处理恶意行为。

2.2　Attacher 和 CTable

构件被调用时会把自身的 8 位 CID 拼接在分组临时头部的 WAID 的尾部，这个工作是 Attacher 完成的。另外，Attacher 还有 2 个任务：一个是 8 位 CW 或 EW 的值加入 AW，另一个是检查是同样的 CIDs 存在于同一个 WAID。如果遇到同样的 CID，Attacher 会执行去除环的操作，最后把增值的 CID 拼接入 WAID。

Ctable 用来给每个构件分配 CW 和 CID。CW 和 CID 的值是从正常 AID 的集合计算出来的。当构件提交自己的信息到 CM 时，Ctable 就会分配对应的 CTAG 给这个构件。

2.3　验证模块 Checker 和 RAT

Checker 用来验证是否当前 AID 是正常行为，实现方法见节 1.3 和 1.4。RAT 使用位图结构存储和查找目标。RAT 的地址形式是 AW-CID-CID，长 24 b。RAT 中的条目占 1 位，值为 1 表明正常，值为 0 是恶意。因此所有 RAT 占用了 2^{24} b 即 2 MB 的空间。RAT 在路由器运行时仅仅可读，以此防止被恶意修改。RAT 的内容根据正常 WAID 计算得来。

2.4　Disabler 处理恶意行为和生成 RAT

Disabler 被放在 SendPacket 构件。当发现恶意行为时，分组被释放，并记录日志信息。在生成 RAT 前，对所有构件遍历并递增编码 CID，按照管理员定义生成正常 AID 集合，如图 4 左边的 AID Set。

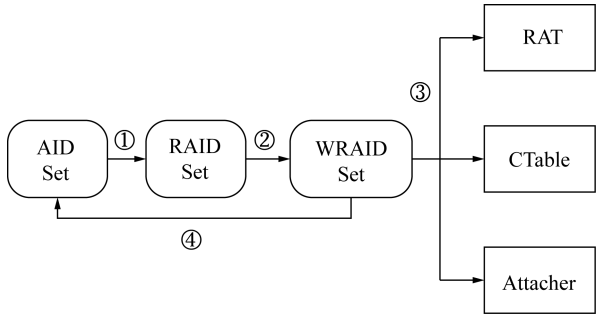


图 4　初始化 RAT

图 4 中，对 AID 集合生成 RAID，使用节 1.4 的方法去除行为中的环。其次基于 RAID，使用算法分配 CW 和 EW，生成 WRAID。分配算法是图遍历算法，算法目标是为每个 CID 计算 CW 和 EW。算法遍历 RAID 所有节点，找到出度大于 1 的节点，调整合适的 CW 和 EW 改变冲突行为的 AW 直到不冲突。最后，使用 WRIAD 配置 RAT、Ctable 和 Attacher。

当有新的行为需要更新 RAT 时，见图 4 的步骤 4。将新的 WRAID 恢复重复的 CID 并去除 AW，生成对应的 AID。这个步骤是步骤 1 和 2 的逆过程，在节 1.4 中已经证明去环方法可逆。

3 范式路由器的效果评价

Click 范式路由器部署在 64 位 Ubuntu 12.04 版本上，配备 8 核 Intel Core i7-3770 CPU、8 GB DDR3 内存和 2 个 100 Mb/s RTL8139 以太网适配器。DPDK 范式路由器部署在 64 位 Ubuntu 14.04.3 版本上，带有 8 核 Intel Core i7-4790K CPU、32 GB DDR3 内存和 4 个 Intel I350-T4 Gb/s 网卡。

Click 和 DPDK 路由器执行基本的路由功能。在 Click 路由器的实验中，5 MB 和 60 kB 的数据在 2 个场景下传输：TCP 场景、TCP 和 UDP 混合的场景。在 DPDK 路由器实验中，路由器直接连接了 Spirent TestCenter 发包机(packet generator)[18]，发包机使用 TCP 进行吞吐率测试。

3.1 范式路由器的有效性

为了调查 Click 范式路由器的有效性，本文在转发构件 SendPacket 中加入了 3 种恶意调用：复制分组、读取分组和随机丢弃分组(丢包概率为 0.1)。每种调用运行在 2 种传输环境：TCP 场景、TCP 和 UDP 混合的场景。本文统计了范式路由器捕获的所有行为的次数。范式路由器对于复制分组和读取分组 2 个恶意调用的表现相似，故读取分组的图忽略不画。图 5 的纵坐标每个值代表 1 种行为类型，由于行为功能多样，纵坐标略去行为名称，以数字代表行为，但图 5a 和 5b 的纵坐标不相关。

在图 5a 中，只要正常行为调用了 SendPacket 构件，就会触发对应的恶意复制行为。在图 5a 中 0～10 ms，路由器会执行寻找地址等正常行为，这些行为只有少数会调用 SendPacket，故触发了少数对应的恶意复制行为。图 5a 中，10～50 ms 时间内，路由器同一个路径的 2 个方向分别执行转发行为，对应图下方 2 种正常行为。它们触发了对应的 2 种恶意复制行为，对应图上方 2 种恶意行为。由于恶意复制行为并行于正常行为，它们并不影响正常行为。图 5b 中，恶意行为发生的密度较小，因为恶意丢包调用发生的概率是 0.1，并且它们会影响正常行为。在图 b 中，TCP 由于重传机制而传输成功而 UDP 没有传输成功，因此诱发了比图 5a 要多的正常行为类型。

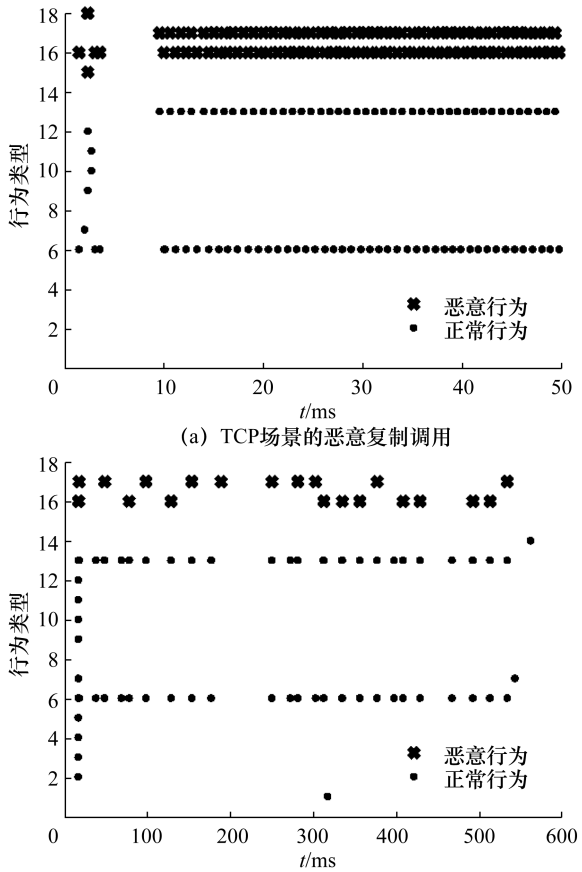


图 5 行为时间点

3.2 范式路由器的性能

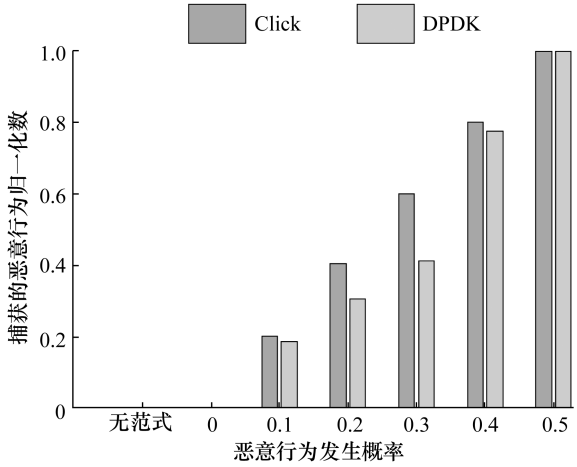
对于 Click 范式路由器，本文使用 iperf 测量吞吐率[19]。对于 DPDK，本文使用 Spirent TestCenter 发包机，配置 TCP 协议来测量吞吐率。对于 2 种路由器，分别都有 7 个实验：无范式路由器，有范式路由器且恶意行为发生概率分别为 0、0.1、0.2、0.3、0.4、0.5。本文得到了恶意行为的归一化数量以及归一化吞吐率，如图 6 所示。

在图 6a 中，由范式路由器捕获的恶意行为数量基本正比于恶意行为的发生概率。在 Click 和 DPDK 路由器上的恶意行为的最大次数是 55 572 和 38 400。Click 部署了范式路由器涉及的构件更多，导致了比 DPDK 更多的行为。

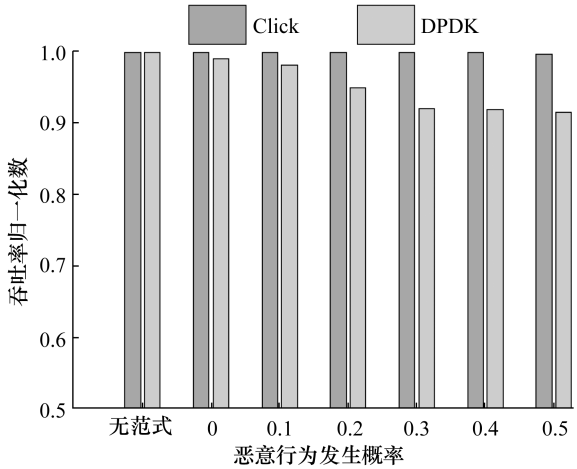
在图 6b 中，归一化的吞吐率会随着恶意行为的增多而轻微减小。主要是因为范式路由器需要将恶意行为信息写入文件，这个 I/O 操作导致了路由器的性能下降。Click 范式路由器的吞吐率被影响的程度比 DPDK 范式路由器被影响的程度要小。因为 Click 是 100 Mb/s 带宽网卡，比 DPDK 范式路由器的 1 Gb/s 网卡的带宽要小很多，影响也会较小。Click 和 DPDK 的最小、最大吞吐率分别是 92.0、93.1 Mb/s 和 860.9、940.2 Mb/s。

4 结 论

本文提出了范式验证机制来保证路由器在动态运行时的行为可信性。基于对路由器数据层的行为的观察，以及恶意行为的特点，设计了一系列方法来监控动态行为。为了降低对转发性能的影响，范式路由器采用了分而治之的验证思想，并使用了位



(a) 范式路由器捕获的恶意行为数量



(b) 范式路由器的吞吐率

图 6　范式路由器随恶意行为增多时的性能

图结构。为了提高验证有效性，范式路由器使用了权重和去除有向环的思想。范式路由器部署在 Click 和 DPDK 路由器环境，评价结果显示范式机制在动态数据层捕获了恶意行为，同时也满足了路由器的性能和代价的要求。

参考文献　(References)

[1] YANG T, XIE G G, LI Y B, et al. Guarantee IP lookup performance with FIB explosion [C]// Proceedings of the 2014 ACM SIGCOMM Conference. Chicago, USA: ACM, 2014: 39 - 50.

[2] APPENZELLER G, KESLASSY I, MCKEOWN N. Sizing router buffers [C]// Proceedings of the 2004 ACM SIGCOMM Conference. Portland, USA: ACM, 2004: 281 - 292.

[3] MALTZ D A, XIE G, ZHAN J, et al. Routing design in operational networks: A look from the inside [C]// Proceedings of the 2004 ACM SIGCOMM Conference. Portland, USA: ACM, 2004: 27 - 40.

[4] PAXSON V. End-to-end routing behavior in the Internet [J]. IEEE/ACM Transactions on Networking, 1997, 5(5): 601 - 615.

[5] XU K, CHEN W L, LIN C, et al. Towards practical reconfigurable router: A software component development approach [J]. IEEE Network, 2014, 28(5): 74 - 80.

[6] NSA/CSSM 1 - 52. Prism project [Z/OL]. [2018-01-15]. https://nsa.gov1.info/dni/prism.html.

[7] MALTZ D A, ZHAN J, XIE G, et al. Structure preserving anonymization of router configuration data [C]// Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. Taormina: ACM, 2004: 239 - 244.

[8] ZHANG Y, PAXSON V. Detecting backdoors [C]// Proceedings of the 9th Conference on USENIX Security Symposium. Denver: USENIX Association Berkeley, 2000: 12 - 12.

[9] SPARKS S, EMBLETON S, ZOU C C. A chipset level network backdoor: Bypassing host-based firewall & IDS [C]// Proceedings of the 4th International Symposium on Information, Computer, and Communications Security. Sydney: ACM, 2009: 125 - 134.

[10] MOJO66. Backdoor found in arcadyan-based Wi-Fi routers [Z/OL]. [2018-01-15]. http://it.slashdot.org/story/12/04/26/1411229/backdoor-found-in-arcadyan-based-wifi-routers.

[11] JC. RuggedCom-backdoor accounts in my SCADA network? you don't say… [Z/OL]. [2018-01-15]. http://seclists.org/fulldisclosure/2012/Apr/277.

[12] COSTIN A, ZADDACH J, FRANCILLON A, et al. A Large-scale analysis of the security of embedded firmwares [C]// Proceedings of the 23rd USENIX Conference on Security Symposium. San Diego: USENIX Association Berkeley, 2014: 95 - 110.

[13] GOODIN D. Malicious cisco router backdoor found on 79 more devices, 25 in the US [Z/OL]. [2018-01-15]. https://arstechnica.com/information-technology/2015/09/malicious-cisco-router-backdoor-found-on-79-more-devices-25-in-the-us/.

[14] GOODIN D. Cisco routers in at least 4 countries infected by highly stealthy backdoor [Z/OL]. [2018-01-15]. https://arstechnica.com/information-technology/2015/09/attackers-install-highly-stealthy-backdoors-in-cisco-routers/.

[15] HIGGINS P, KRISHNAN R. DEFCON router hacking contest reveals 15 major vulnerabilities [Z/OL]. [2018-01-15]. https://www.eff.org/ru/node/82002.

[16] DOBRESCU M, ARGYRAKI K. Software dataplane verification [C]// Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation. Seattle: USENIX Association Berkeley, 2014: 101 - 114.

[17] KIM T H, BASESCU C, JIA L, et al. Lightweight source authentication and path validation [C]// ACM SIGCOMM. Chicago: ACM New York, 2014: 271 - 282.

[18] Spirent. Spirent packet generator [Z/OL]. [2018-01-15]. http://www.spirent.com/Products/TestCenter.

[19] DUGA J, ELLIOTT S, MAH B A, et al. Iperf [Z/OL]. [2018-01-15]. https://iperf.fr/.

[20] ABADI M, BUDIU M, ERLINGSSON U, et al. Control-flow integrity [C]// Proceedings of the 12th ACM Conference on Computer and Communications Security. Alexandria: ACM New York, 2005: 340 - 353.

(责任编辑　刘森)