基于双层信息流控制的云敏感数据安全增强

吴泽智[1,2],陈性元[1,2],杜学绘[1],杨 智[1]
(1. 信息工程大学密码工程学院,河南郑州 450001;2. 密码科学技术国家重点实验室,北京 100094)

摘 要: 已有的云安全防护方法如加密、访问控制和虚拟机隔离等不能够提供数据端到端的安全防护。首先,提出了一个面向云环境的双层信息流控制模型,给出了模型的关键要素定义、集中式与分布式信息流控制规则、能力标记调整规则、标记传播规则和降密规则. 然后,综合动态污点跟踪和虚拟机自省技术,设计并实现了原型系统 IFCloud,可为云租户提供信息流跟踪与控制即服务,为云平台提供常见系统攻击如栈溢出、缓冲区溢出等攻击的防护机制. 最后,给出了原型系统 IFCloud 的功能测试结果. 表明 IFCloud 能够灵活、正确、实时地跟踪和控制云下敏感数据流. 可应用于云平台下面向软件即服务的细粒度数据安全保护.

关键词: 云数据安全; 信息流控制模型; 动态污点跟踪; 虚拟机自省; 栈溢出攻击
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112 (2018)09-2245-06
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.09.028

Enhancing Sensitive Data Security Based-on Double-Layer Information Flow Controlling in the Cloud

WU Ze-zhi[1,2], CHEN Xing-yuan[1,2], DU Xue-hui[1], YANG Zhi[1]
(1. *College of Cryptogram Engineering, PLA Information Engineering University, Zhengzhou, Henan* 450001, *China*;
2. *State Key Laboratory of Cryptology, Beijing* 100094, *China*)

Abstract: The existing security methods in the cloud such as encryption, access control, and VM isolation can not guarantee end-to-end data security. To address this problem, a double-layer information flow control model is proposed. The definition of key element, centralized and decentralized information flow rules, capability adjustment rules, label propagation rules, and declassification rules of the model are presented. Then, taking the advantages of dynamic taint tracking and virtual machine introspection technologies, a prototype system named IFCloud are designed and implemented. IFCloud achieves information flow tracking and controlling as a service for cloud tenant and provides detection methods against common system attacks such as stack and buffer overflow attack for the cloud provider. Finally, IFCloud is demonstrated to be a flexible and accurate system that tracks and controls the sensitive data flow in the cloud at runtime from the function test results, and it can be applied to protect data security at a fine-grained level for the software as a service cloud.

Key words: cloud data security; information flow control model; dynamic taint tracking; virtual machine introspection; stack overflow attack

1 引言

云计算在方便人们生活的同时也带来了严重的敏感数据安全问题. 文献[1]研究了云数据完整性验证方案. 文献[2]综述了基于属性的云数据安全,文献[3]介绍了云应用安全相关研究进展. 但上述工作都不能为云计算提供端到端数据安全保障. CloudSafetyNet[4]基于网络通信保证了云租户之间数据与隐私安全. 但 CloudSafetyNet 缺乏云租户虚拟机内数据安全机制. CloudFence[5]基于 pin 插装平台实现了云下敏感数据流的跟踪,有效保证了云中租户数据隔离与安全共享,但其缺乏有效的信息流控制模型,不支持灵活可用的信息流控制策略. FlowK[6]和 FlowR[7]在操作系统层实现了进程级别粗粒度的信息流控制,不能提供细粒度的信息流控制机制. 文献[8]实现云下基于信息流的审计. 文献[9]将可信平台技术与分散式信息流控制技术相结

收稿日期:2017-08-29;修回日期:2017-12-28;责任编辑:覃怀银
基金项目:国家高技术研究发展计划(863)(No. 2015AA016006, No. 2012AA012704)

合并应用云数据安全增强. 以上工作仅在租户层面考虑数据安全,未考虑为云服务提供商提供保护机制. 综上,有必要研究细粒度的信息流控制机制,对于云租户,既能够保证云租户虚拟机内部数据安全,又能够保证云租户之间的数据安全与共享. 对于云服务提供商,能够保护云平台不受恶意租户攻击.

2 双层信息流控制模型

2.1 模型安全标签设计

信息流控制机制实现的核心思想是将标签附着在数据上,标签随着数据在整个系统中传播,即数据派生出的对象也将会继承原有数据标签. 利用这些数据标签能够限制程序间的数据流向. 本文安全标签设计如下:

定义 1 基础标签 l_i 表示某一类型的敏感数据或恶意数据类型. 在机密域,某一种基础标签可表示某一类秘密数据. 在完整域,某一种基础标签可表示某一类恶意数据. 标签 L_i 由不同种 l_i 组成集合,代表某一类或者多类敏感数据或恶意数据类型.

定义 2 标签格 表示为$(L,\wedge,\perp,T)$,标签值域是一个乘积格. L 表示所有标签的集合,对于任意标签值 x 属于标签值域 L,交汇运算 $\wedge$ 取集合并集 $\cup$,其顶元素是空集$\varnothing$,表示为 T. 交汇运算符定义了标签值域上的一个偏序(记为$\leqslant$),若有 $L_1\leqslant L_2$,$L_2\leqslant L_1$ 则有 $L_1=L_2$.

定义 3 反转标签格 表示为 L_{re},定义为:反转标签格的顶元素是标签格的底元素 $\perp_{re}=T$,反转标签格的底元素是标签格的顶元素 $T_{re}=\perp$,同理,通过对称方法可求反转标签格中其它元素值. 其有如下性质:反转标签的值域仍然是原标签格值域 $L_1=L_2$,且 $L_1\leqslant_{re}L_2\Leftrightarrow L_2\leqslant L_1$. 若标签格的偏序性质用于机密性保护,则反转标签格偏序性质可用于完整性保护.

定义 4 标签格卡尔笛积 表示为 $L_\times=L_c\times L_i$.

安全系统通常需要机密性和完整性保护,因而需要统一机密性标签和完整性标签. 标签格卡尔笛积有如下性质:$(L_{c1}\times L_{i1})\leqslant(L_{c2}\times L_{i2})=((L_{c1}\leqslant L_{c2})\wedge(L_{i1}\leqslant L_{i2}))$,若机密性标签格满足偏序关系且完整性标签格满足偏序关系,则标签格卡尔笛积满足偏序关系.

定义 5 标签互斥 标签互斥是指在需要约束的标签集 R 中,对于任意保密性或完整性标签 X,需满足 $|R\cap X|<2$. 即标签 X 不能同时具备互斥约束集 R 中两个或以上的基础标签.

通过标签互斥可有效地实现最小特权原则和职责分离原则. 制定最小特权原则时,仅给该主体标签分配相应的基础标签;制定职责分离原则时,将不同的基础标签分配给不同主体标签.

2.2 集中式信息流控制

集中式信息流控制主要元素和规则如图 1 所示,主要包括进程、数据、标记、操作四个要素和信息流发送和接收控制规则、进程创建规则与标记传播规则.

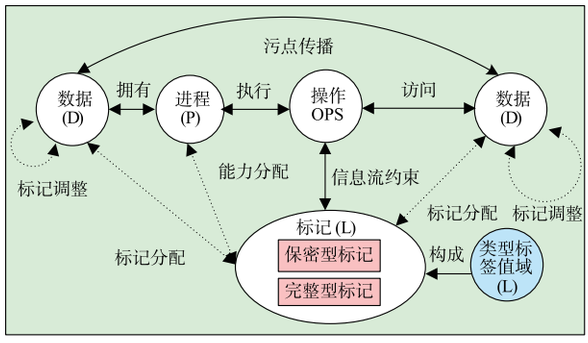

图1 集中式信息流控制示意图

规则 1 信息流发送控制规则

$$A\xrightarrow{S}D \text{ if } (\mathrm{SL}_C(D)\times\mathrm{SL}_I(D))\leqslant(\mathrm{SL}_C(A)\times\mathrm{SL}_I(A)) \tag{1}$$

式(1)表示进程 A 发送数据 D 的必要条件是:数据 D 的标记与进程 A 的发送能力标记满足偏序关系.

规则 2 信息流接收控制规则

$$A\xrightarrow{R}D \text{ if } (\mathrm{RL}_C(D)\times\mathrm{RL}_I(D))\leqslant(\mathrm{RL}_C(A)\times\mathrm{RL}_I(A)) \tag{2}$$

式(2)表示进程 A 接收数据 D 的必要条件是:数据 D 的标记与进程 A 的接收能力标记满足偏序关系.

规则 3 进程创建规则

$$\text{if } A\xrightarrow{\text{copy}}B \text{ then } (L_C(B)=L_C(A)) \text{ and } (L_I(B)=L_I(A)) \tag{3}$$

式(3)表示通过复制方式进程 A 创建进程 B 后,进程 B 安全标记继承进程 A 安全标记. 例如:通过 fork 创建的进程可采用该规则标记新进程.

$$\text{if } A\xrightarrow{\text{exe}}B \text{ then}(L_C(B)=P(B)) \text{and}(L_I(B)=P(B)) \tag{4}$$

式(4)表示通过执行方式进程 A 创建进程 B 后,进程 B 安全标记由其策略文件 P(Policy)决定. 例如:通过 exec 加载的特定的可执行二进制程序时,可使用该规则标记新创建的进程.

规则 4 标记传播规则

$$\begin{aligned}&\text{if } D_a\rightarrow D_b \text{ then } (L_C(D_b)=L_C(D_a)\cup L_C(D_b))\\&\quad\text{and}(L_I(D_b)=L_I(D_a)\cup L_I(D_b))\end{aligned} \tag{5}$$

式(5)表示信息流从数据 D_a 流向 D_b 时,数据 D_b 的标记更新为 D_a 与 D_b 标记相并(标记格内交汇运算).

标记传播分为两种情形:第一,进程执行过程中内部的标记传播. 例如,程序执行赋值指令 $b=a$ 时,信息流从数据 a 流向了数据 b,需要将 b 的标记进行更新. 第二,进程之间通信产生的标记传播. 例如,进程 A 向进程 B 通过套接字发送了信息 a,进程 B 接收该信息并保存

在数据 b 中,此时信息流从数据 a 流向了数据 b,需要将 b 的标记进行更新.

2.3 分布式信息流控制

分布式信息流控制元素及规则如图 2 所示,其主要包括租户、数据、标记、能力、操作五个要素和信息流发送和接收控制规则、标记传播规则、能力调整规则和降密规则. 与集中式信息流控制不同,分布式信息流控制还设计了租户的能力,能力表示租户对自身标记的调整能力. 能力调整规则和数据降密规则如下:

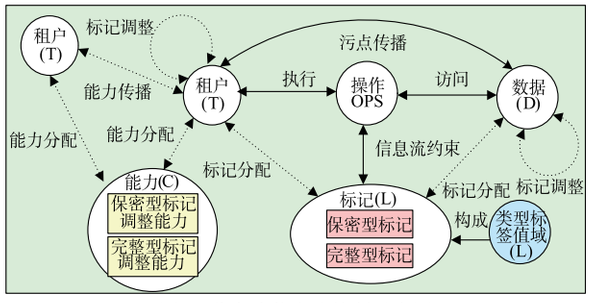

图2　分布式信息流控制示意图

规则 5　能力调整规则

$$L_C(A) = L_C(A) \cup \{l_i\} \text{ if } l_i \in P_C^+(A) \tag{6}$$

式(6)表示租户 A 可将标记 l_i 加入到自身机密性标记中的必要条件是:标记 l_i 属于租户特权集 $T_C^+(A)$.

$$L_I(A) = L_I(A) \cup \{l_i\} \text{ if } l_i \in P_I^+(A) \tag{7}$$

式(7)表示租户 A 可将标记 l_i 加入到自身完整性标记中的必要条件是:标记 l_i 属于租户特权集 $T_I^+(A)$.

规则 6　数据降密规则

$$L_C(D) = L_C(D) \setminus \{l_i\} \text{ if } l_i \in T_C^-(A) \tag{8}$$

式(8)表示租户 A 可将数据机密性标记 l_i 删除的必要条件是:标记 l_i 属于租户降密能力集 $T_C^-(A)$. 例如:敏感数据由于时间等原因可能不再是敏感数据,需要将其敏感标记删除.

$$L_I(D) = L_I(D) \setminus \{l_i\} \text{ if } l_i \in T_I^-(A) \tag{9}$$

式(9)表示进程 A 可将数据完整性标记 l_i 删除的必要条件是:标记 l_i 属于租户降密能力集 $T_I^-(A)$.

2.4 双层信息流控制模型特征

综上,集中式与分布式信息流控制的区别主要体现在信息流控制策略由谁制定. 集中式信息流控制策略完全由管理员或者租户制定. 例如,云租户虚拟机内部,租户运行众多应用程序,各应用程序具备不同的数据处理权限,何种应用具备何种权限是由云租户决定的,也可以说策略是集中制定的. 分布式信息流控制策略是由不同租户共同制定的. 例如,云租户 A 可制定与云租户 B 或租户 C 的信息流控制策略,但云租户 A 不能够制定云租户 B 与租户 C 之间的信息流控制策略,即与其它云租户通信或共享数据时,各云租户仅制定与自身相关的安全策略,由这些策略构成一个分布式策略集合. 总体上,模型实施了以数据安全为中心的集中式与分布式信息流控制策略,通过引入安全标记互斥与约束,可以更好支持最小特权和职责分离原则,通过能力调整表达无中心的分布式降密.

3 设计与实现

综合已有的动态污点跟踪技术[10]、虚拟机自省技术(Virtual Machine Introspection,VMI)[11]和信息流控制技术[12],本文实现了云环境下多租户、多维度、多粒度、全程化、虚拟化、透明化、分布式、动态的信息流跟踪与控制系统. 在云租户虚拟机内部,可提供程序指令级别和操作系统进程通信级别的信息流跟踪和控制机制;在云租户虚拟机之间,可提供网络通信数据字节级别的信息流跟踪和控制机制.

从租户视角出发,原型系统 IFCloud 整体结构如图 3 所示. 云租户 A 和 B 所有程序运行在安装了相应客户操作系统的虚拟机中. 虚拟机主要增加的模块包括二进制级别细粒度污点跟踪模块、即时虚拟机自省模块、集中式与分布式信息流控制模块、审计数据库模块和用户接口模块. 在租户操作系统实现了指令调用级别、函数调用级别和进程通信级别三种类型的基于事件驱动的集中式信息流控制. 各租户间采用分布式信息流控制方法,云租户可通过程序接口制定相应的信息流控制策略和查看信息流审计信息. 综上,IFCloud 实现了云环境下信息流跟踪与控制即服务(Information Flow Tracking and Controlling as a Service,IFTCaaS).

从云服务提供商(云平台)视角出发,原型系统 IFCloud 整体结构如图 4 所示. 云租户所有程序都在云平台提供的虚拟机内隔离运行,并受到监督和控制. 云平台将所有租户的应用程序均看作不可信程序,在程序执行过程中采用动态污点跟踪技术、指令级钩子注入技术、函数级钩子注入技术、API 级钩子注入技术、路径覆盖技术和事件记录技术为动态分析插件接口提供所需的分析信息. 通过实时注册动态插件,可实现异常指令序列检测、异常跳转与方法调用检测、异常 API 调用检测、冷路径检测与分析和网络状态与文件系统状态异常检测等功能. 并将检测结果进行审计和报告. 综上,IFCloud 为云服务提供商提供保护机制,防止恶意租户对云平台攻击和破坏. 可在一定程度上防御栈溢出攻击、缓冲区溢出攻击、SQL 注入攻击、格式化字符串攻击、面向返回的编程(Return-Oriented Programming,ROP)攻击和面向跳转的编程(Jump-Oriented Programming,JOP)攻击等.

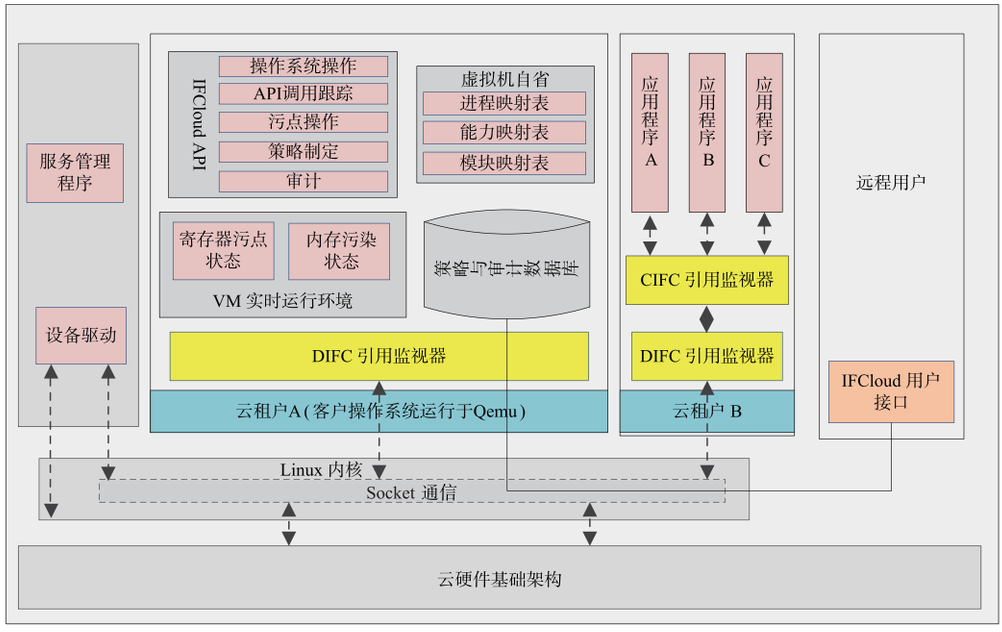

图3 租户视角下IFCloud整体结构

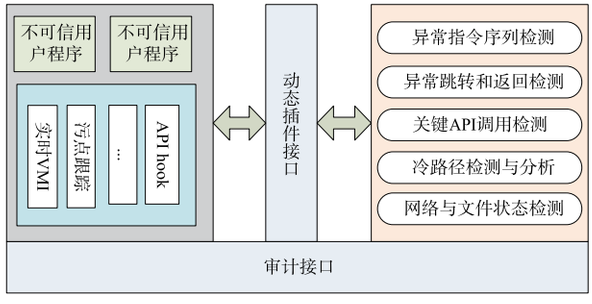

图4 云服务提供商视角下IFCloud整体结构

4 功能测试

测试使用的主机操作系统环境为 OS/Ubuntu-12.04,CPU/2.27GHz,RAM/4GB,云租户操作系统环境为 Ubuntu 9.04 和 Win 7. 首先,在云租户 Ubuntu 9.04 和 Win 7 操作系统下测试系统指令级别和进程通信级别污点跟踪正确性和集中式信息流控制策略实施的正确性. 在 Ubuntu 9.04 环境下配置 gedit 程序机密性标记为 0x00,测试使用键盘输入污点数据,通过 IFCLOUD_TAINTMEM_READ,IFCLOUD_TAINTMEM_WRITE 回调函数记录污点在系统内流动情况. 当数据流入 gedit 程序时,IFCloud 依据策略判断该信息流非法,阻止该操作并审计. 与预期结果一致. 在 Win 7 环境下配置 notepad.exe 程序机密性标记为 0x1000,待测试文件机密性标记为 0x1000. 并通过 IFCloud_Apihook_Add 注册 ntdll.dll 中 NtOpenFile,NtReadFile 和 NtWriteFile 的钩子函数. 测试使用 notepad.exe 程序打开测试文件,IFCloud 依据策略判断该信息流合法,notepad.exe 成功打开该文件并根据文件句柄读文件,与预期结果一致. 测试使用键盘输入污点数据并保存到文件时,IFCloud 检测到该数据含有污点,依据策略判断该信息流合法. 并更新文件污点信息,与预期结果一致.

其次,测试多租户云环境下污点跟踪正确性和分布式信息流控制策略实施的正确性. 其策略配置如表 1 所示,租户 A 制定自身策略分布式标记为[S:B:0x01;R:B:0x01]. [S:B:0x01]表示允许向标识为 B 的租户发送包含 0x01 污点标记的数据. [R:B:0x01]表示允许接收来自标识为 B 的租户包含 0x01 污点标记的数据. [D:A:0x01]表示允许租户 A 清除数据中 0x01 污点标记.

表 1 分布式信息流控制策略配置表

租户	发送能力	接收能力	降密能力
A	B:0x01	B:0x01	A:0x01
B	A:0x01 C:0x01	A:0x01	–
C	–	–	–

租户 A 向租户 B 发送含有污点 0x0001 的数据. 系统通过回调函数 IFCloud_NIC_SEND 获取该次发送对象的租户标识和数据污点信息,依据租户 A 的策略允许该发送动作,并将数据污点值写入通信数据包中;当租户 B 收到该数据包时,通过回调函数 IFCLOUD_NIC_RECEIVE 获取该数据包的来源和污点信息,依据租户 B 的策略允许该接收动作,并将该污点值通过 IF-

CLOUD_TAINT_MEM 接口标记相应内存为污点,测试结果符合预期. 租户 *A* 向租户 *C* 发送含有污点 0x0001 的数据,在租户 *A* 向外发送网络数据包时,依据租户 *A* 的策略禁止该发送动作,测试结果符合预期. 租户 *A* 将含有污点 0x0001 的数据通过自身降密处理后发送给租户 *C*. 租户 *C* 成功收到该数据包.

最后,测试 IFCloud 检测常见系统攻击的效果. 第一,在租户 Win 7 操作系统内进行了缓冲区攻击测试. 测试程序试图利用 strcpy 函数将 char buf1[200]复制到 char buf2[20]. IFCloud 通过注册钩子函数 (IFCloud_hook_function_byname("ntdll.dll","strcpy",1,target-cr3,strcpy_call,NULL,0)截获系统 strcpy 函数调用,当检测到被复制字符串长于复制字符串时,IFCloud 发出缓冲区攻击警告. 第二,在租户 Ubuntu 9.04 操作系统内进行了 ROP 攻击测试. 测试采用 Jonathan Salwan 编写的 ROPgadget 搜索到 12 个可用 gadgets 并组装出攻击程序,目的是通过执行 exec 系统调用打开一个 shell 终端. 使用指令级 ROP 攻击检测插件核心代码如下所示:

```
static void INSN_call(insn[20])
{    int i,count=0;
     for(i=0,i<20,i++)
       {if(insn[i].opcode=ret)
            count++;
       }
     if(count>threshold)
         IFCloud_audit("ROP detected\n");
}
```

IFCloud 系统检测到该攻击并发出警告.

5　结论

通过分析云环境下数据安全需求及当前研究工作存在的不足,综合污点跟踪技术与虚拟机自省技术,构建了云环境下多维度、多粒度的动态信息流跟踪与控制系统,为云租户和云平台提供了一体化安全防护. 功能测试结果表明 IFCloud 能够灵活、正确和实时地跟踪和控制云环境下敏感数据流. 可应用于云平台下面向软件即服务的细粒度数据安全保护.

参考文献

[1] 周恩光,李舟军,郭华,等. 一个改进的云存储数据完整性验证方案[J]. 电子学报,2014,42(1):150 – 154.
ZHOU En-guang, LI Zhou-jun, GUO Hua, et al. An improved data integrity verification scheme in cloud storage system[J]. Acta Electronica Sinica, 2014, 42(1): 150 – 154. (in Chinese)

[2] 王小明,付红,张立臣. 基于属性的访问控制研究进展[J]. 电子学报,2010,38(7):1660 – 1667.
WANG Xiao-ming, FU Hong, ZHANG Li-chen. Research progress on attribute-based access control[J]. Acta Electronica Sinica, 2010, 38(7): 1660 – 1667. (in Chinese)

[3] 肖玮,陈性元,包义保. 可重构信息安全系统研究综述[J]. 电子学报,2017,45(5):1240 – 1248.
XIAO Wei, CHEN Xing-yuan, BAO Yi-bao. Review of research on reconfigurable information security system[J]. Acta Electronica Sinica, 2017, 45(5): 1240 – 1248. (in Chinese)

[4] PRIEBE C, MUTHUKUMARAN D, O'KEEFFE D, et al. CloudSafetyNet: detecting data leakage between cloud tenants[A]. Proceedings of the 6th ACM Workshop on Cloud Computing Security[C]. USA: ACM, 2014. 117 – 128.

[5] PAPPAS V, KEMERLIS V P, ZAVOU A, et al. CloudFence: data flow tracking as a cloud service[A]. Proceedings in Attacks, Intrusions, and Defenses[C]. Berlin: Springer, 2013. 411 – 431.

[6] PASQUIER T F J M, SINGH J, BACON J, et al. An information flow control model for the cloud[A]. International Conference on Cloud Computing Technology and Science[C]. USA: ACM, 2016. 70 – 77.

[7] PASQUIER T F J M, BACON J, SHAND B. FlowR: aspect oriented programming for information flow control in ruby[A]. Proceedings of the 13th International Conference on Modularity[C]. USA: ACM, 2014. 37 – 48.

[8] PASQUIER T F J M, SINGH J, BACON J, et al. Information flow audit for PaaS clouds[A]. IEEE International Conference on Cloud Engineering[C]. USA: IEEE, 2016. 42 – 51.

[9] PASQUIER J M, SINGH J, BACON J. Clouds of things need information flow control with hardware roots of trust[A]. IEEE, International Conference on Cloud Computing Technology and Science[C]. USA: IEEE, 2016. 467 – 470.

[10] HENDERSON A, PRAKASH A, YAN L K, et al. Make it work, make it right, make it fast: building a platform-neutral whole-system dynamic binary analysis platform[A]. ISSTA[C]. USA: ACM, 2014. 248 – 258.

[11] 李保珲,徐克付,张鹏,等. 虚拟机自省技术研究与应用进展[J]. 软件学报,2016,27(6):1384 – 1401.
LI Bao-hui, XU Ke-hu, ZHANG Peng, et al. Research and application progress of virtual machine introspection technology[J]. Journal of Software, 2016, 27(6): 1384 – 1401. (in Chinese)

[12] 吴泽智,陈性元,杨智,等. 信息流控制研究进展[J]. 软件学报,2017,28(1):135 – 159.
WU Ze-zhi, CHEN Xing-yuan, YANG Zhi et al. Survey

on information flow control [J]. Journal of Software, 2017,28(1):135 – 159. (in Chinese)

作者简介

吴泽智 男,1990 年生于湖南长沙. 现为信息工程大学密码工程学院博士研究生. 主要研究方向为信息流控制与云计算安全.
E-mail:1141208772@qq.com

陈性元 男,1963 年生于安徽无为. 现为信息工程大学教授、博士生导师. 主要研究方向为网络与信息安全,大数据安全等.
E-mail:chxy302@vip.sina.com

杜学绘(通信作者) 男,1968 年生于河南辉县. 现为信息工程大学教授、博士生导师. 主要研究方向为空天网络安全,云计算与大数据安全.
E-mail:dxh37139@sina.com

杨 智 男,1975 年生于河南开封. 现为信息工程大学副教授、硕士生导师. 主要研究方向为信息流控制,操作系统安全与云计算安全.
E-mail:zynoah@163.com