

基于操作系统多样性的虚拟机安全部署策略

张淼, 季新生, 艾健健, 刘文彦, 扈红超, 霍树民

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

摘要: 云计算的资源共享模式, 在极大提高资源利用率的同时, 也带来了诸多安全问题, 如虚拟机间的共存攻击。特别是当用户使用单一的操作系统时, 攻击者可以在较小的开销下, 攻破用户的全部虚拟机, 从而窃取隐私和数据。针对这一安全威胁, 利用不同操作系统漏洞存在差异的特点, 提出一种基于操作系统多样性的虚拟机安全部署策略。该方法首先为申请虚拟机的用户推荐一种多样性程度最高的操作系统配置选择; 然后通过一种安全的部署策略, 最大化地发挥多样性的特点, 使攻击者需要付出更大的开销。实验结果表明, 与单一操作系统配置的方法相比, 该方法至少可以降低 33.46% 的攻击效益。

关键词: 虚拟机; 共存攻击; 操作系统; 多样性; 云安全

中图分类号: TP302

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2017.00201

Secure deployment strategy of virtual machines based on operating system diversity

ZHANG Miao, JI Xin-sheng, AI Jian-jian, LIU Wen-yan, HU Hong-chao, HUO Shu-min

(National Digital Switching Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract: The resource-sharing model for cloud computing raises many security issues, such as co-resident of virtual machines, while greatly improving resource utilization. In particular, when a user adopts a single operating system, an attacker can steal privacy and data by compromising the user's entire virtual machine at a smaller cost. In view of this security threat, a strategy for the security deployment of virtual machine based on operating system diversity was presented. This method firstly recommended an operating system configuration options for users applying for virtual machines with the highest degree in diversity, and then through the secure deployment strategy, maximized the effect of diversity, thus making the attacker pay more cost. The experimental results show that compared with the method of single operating system, this method can reduce the attack efficiency by 33.46% at least.

Key words: virtual machine, co-resident attack, operating system, diversity, cloud security

1 引言

云计算是一种新型的模式, 它向广大用户提

供透明的分布式服务, 使用户可以共享各种底层物理资源。作为能够高效提供大规模存储和计算能力的技术, 云计算得到了广泛的关注, 其中,

收稿日期: 2017-07-11; 修回日期: 2017-09-01。通信作者: 张淼, 1021380925@qq.com

基金项目: 国家自然科学基金青年基金资助项目 (No.61602509); 国家自然科学基金创新研究群体基金资助项目 (No.61521003); 国家重点研发计划基金资助项目 (No.2016YFB0800100, No.2016YFB0800101)

Foundation Items: The National Science Foundation for Distinguished Young Scholars of China (No.61602509), The Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No.61521003), The National Key R&D Program of China (No.2016YFB0800100, No.2016YFB0800101)

虚拟化是云计算的一个重要特征。虚拟化技术将底层的物理资源抽象化,可以根据用户需求动态分配资源,不仅提高了计算能力,同时可以降低能耗。

为了支持以上的特征,研究人员提出以虚拟机作为云环境的原生构件。为了最大化底层物理平台的使用效率,通常不同用户的虚拟机需要运行在同一服务器上,且彼此之间逻辑隔离开来,共享底层物理资源。虚拟机还可以方便地部署于不同的物理主机或者整合到同一主机上以实现负载均衡或提高使用率。此外,利用虚拟机的固有特性,还可以在进程不中断的情况下实现对虚拟机的迁移。

云计算引入了多租户模式,为资源的高利用率带来了可能。然而,不同用户共享底层物理资源,也会带来一些安全隐患。例如,用户由于个人习惯等其他因素,可能会在申请虚拟机实例时,要求使用同一类型的操作系统(OS, operating system),如 Windows 系统。这样的做法方便了用户处理数据和加载应用,然而带来的问题是,如果用户虚拟机所在的服务器上有共存的恶意攻击者,并且锁定了目标用户,同时知道目标虚拟机的位置,那么攻击者只需要发起一次探测和漏洞挖掘过程,就可以利用漏洞攻击用户的虚拟机,这是一种典型的共存攻击。即使用户的虚拟机分布在云环境中不同的物理主机上,如果所有虚拟机采用相同的操作系统,这样的配置也会给攻击者创造极为便利的攻击条件,导致攻击者很容易获取该用户所有存储的数据和隐私信息,这是非常严重的后果。即使是 Windows 的不同版本(如 Windows7、Windows8),由于大量共同漏洞的存在,仍然给攻击者节省了大量的成本和时间开销,对于正常用户的数据安全形成威胁。事实上,很多操作系统之间由于编程语言、运行环境等差别,是没有共同漏洞存在的。因此,如果在虚拟机的操作系统中引入多样性,必然使攻击者需要付出更大的代价才能窃取信息。本文考虑利用多样性的思想来缓解共存攻击问题。

2 相关工作

云环境中的共存攻击问题自被提出以来,就

不断有研究者从攻、防两方面进行研究;而多样性由于其安全属性,被广泛应用于加强系统可靠性。本节分别介绍先前对这2个内容的相关研究工作。

2.1 共存攻击及其防御方法

为了最大化资源的利用率,云服务提供商(CSP, cloud service provider)通常会将不同用户的虚拟机部署在同一物理主机上,这样的虚拟机之间会形成一种共存(或称之为同驻)关系。2009年,Ristenpart等^[1]提出了共存虚拟机存在的安全问题,即攻击者可以通过发起侧信道攻击从目标虚拟机中窃取信息,如缓存使用和负载信息等。后续对于共存攻击的研究逐渐增多,Apecechea等^[2]在Xen和VMware这样主流的虚拟化工具上实现了细粒度的攻击手段,从目标虚拟机中获取了AES加密密钥。而探测信息的方法也在不断完善,如Osvik等^[3]提出的Prime-Probe方法、Yarom等^[4]提出的Flush-Reload方法等,都可以在不同层面发起攻击,对用户隐私及数据安全造成严重威胁。

针对这种攻击形式,研究者提出了一系列防御方法。例如,文献[5]表明,在操作系统层可以在受保护的进程中加入噪音进行防御;文献[6]提出可以在硬件设计时采用接入随机化和资源分割进行防御。然而这些方法只能够防御特定类型的共存攻击,且需要对底层软硬件进行修改。文献[7]提出,通过虚拟机迁移,减少共存的时间,从而导致攻击者无法成功窃取目标信息。该方法不需要修改底层的软硬件,但随着虚拟机数量的增多,迁移开销也将增大,难以适用于大规模的云数据中心。

2.2 多样性

早在20世纪70年代,就有文献提到利用N版本编程技术创建多样性软件组件^[8]。随着网络安全得到越来越多的关注,利用多样性思想解决这类问题也得到广泛研究。Forrest等^[9,10]将生物学中的概念应用于计算机安全,提出多样性是一种重要的减轻攻击效果的机制;文献[11]在SQL服务器上应用多样性进行实验研究,证明多样性可以提高系统可靠性;Newell等^[12]研究在路由节点上如何分配多样化变体,以增加网络弹性;

Garcia 等^[13]在入侵容忍系统中引入操作系统多样性,提升系统健壮性。事实上,在软硬件、平台、数据等各个层面都可以利用多样性提升系统的安全性,实现特定的安全需求。

基于已有研究可知,系统多样性程度越高,整体的可靠性和安全性也越高。

3 威胁模型和本文目标

3.1 攻击者威胁模型

本文假定 CSP 及其设施和平台都是可信的,也就是说本文不考虑攻击者通过云管理系统的漏洞来破坏云环境的基本功能,从而窃取信息。结合实际情况,在本文的威胁模型中,对攻击者做如下假设。

1) 目标鉴定:攻击者可以判断感兴趣的目标用户是否以及何时与自己的虚拟机共存,如研究用户行为模式或利用外部的探测。即这部分行为对于攻击者的开销较小,可以忽略不计。

2) 完整攻击的能力:一次完整的攻击包括对目标操作系统的识别、系统漏洞或预装软件后门的挖掘以及后续的嗅探和劫持攻击等。攻击者有能力对任何一种操作系统发起完整攻击,但攻击过程会受到资源、开销的约束,即攻击者无法对所有目标虚拟机都实行完整的攻击。

3) 不可预知:用户或 CSP 无法预先认定某一特定用户是否是恶意的攻击者,即无法保证部署的服务器上没有攻击者预先启动的虚拟机。

4) 针对虚拟机的攻击:攻击者每次都直接对用户的虚拟机发起攻击,不考虑攻陷 hypervisor 后,获取所有虚拟机控制权限的情形。

攻击者具备目标鉴定的能力在较多研究中已经提到^[14,15],而发起针对虚拟机的攻击是当前研究的热点^[1,16],成本约束和不可预知性则是云环境的基本特点决定的。因此,上述对于攻击者的假设具有合理性和可研究性。

3.2 本文目标

基于上述对攻击者能力的设定,可以发现,攻击者在开销的约束下,有可能采取只识别、挖掘一次漏洞,而利用该漏洞攻击所有目标的策略,如果没有多样性的操作系统,攻击者就可以以较小的成本窃取信息并迅速将攻击范围扩大。因此,

本文的工作目标是希望能解决以下 2 个具体的现实问题。

1) 正常用户申请若干虚拟机实例时,如何配置这些虚拟机的操作系统,可以使其多样性程度最高,从而带来最高的安全收益。

2) 这些虚拟机在物理主机上如何部署,可以在资源约束下尽可能提高用户虚拟机整体的可靠性和安全性。

4 基于多样性最优的操作系统配置方法

4.1 基本描述

假设用户需要申请 N 个虚拟机,记为 $\{VM_1, VM_2, \dots, VM_N\}$, 用户可以使用的操作系统有 m 种,为 $\{O_1, O_2, \dots, O_m\}$ 。定义一个变量 u_{ij} , 表示第 i 个虚拟机是否使用第 j 种操作系统。即

$$u_{ij} = \begin{cases} 1, & \text{第 } i \text{ 个虚拟机使用第 } j \text{ 种操作系统} \\ 0, & \text{其他} \end{cases} \quad (1)$$

显然,一个虚拟机从创建到销毁的时间段内只能采用一种操作系统,因此存在唯一性约束。

$$\sum_{j=1}^m u_{ij} = 1, i = 1, 2, \dots, N \quad (2)$$

由于 m 是远小于 N 的,为了尽可能使所有虚拟机包含的操作系统种类更多,要求每种操作系统至少被使用一次,即

$$\sum_{i=1}^N u_{ij} \geq 1, j = 1, 2, \dots, m \quad (3)$$

定义操作系统类型 j 的频率为

$$p_j = \frac{\sum_{i=1}^N u_{ij}}{N}, j = 1, 2, \dots, m \quad (4)$$

4.2 多样性模型

对于 m 种操作系统,由于操作系统之间共同漏洞的存在,定义一个相关性矩阵 $\mathbf{X}$, x_{ab} 表示 a 、 b 这 2 种操作系统之间的关联程度,即攻破操作系统 a 后,操作系统 b 被攻破的概率,且 $x_{aa} = 1$ 、 $x_{bb} = 1$ 。本文定义关联程度只由共同漏洞数刻画,共同漏洞数越大,则关联程度越高,对应的元素值更大。设 a 、 b 间共同漏洞数为 K_{ab} , a 本身漏洞数为 V_a , 其中 $K_{ab} = K_{ba}$, 令

$$x_{ab} = \frac{K_{ab}}{V_a}, \forall 1 \leq a, b \leq m$$

本文借鉴生物多样性中基于物种相似度的多样性度量方法^[17], 来评价操作系统之间的多样性。首先, 对于任意操作系统 j , 令

$$w_j = \sum_{a=1}^m x_{ja} p_a, j=1, 2, \dots, m$$

则定义操作系统的多样性度量为

$$E(u) = \frac{1}{\prod_{j=1}^m w_j^{p_j}} \quad (5)$$

式(5)中 u 代表某种操作系统配置方法。本文的目标即可转化为寻求多样性度量 $E(u)$ 取最大值时的操作系统配置方法。即

$$\begin{aligned} \text{Max } E(u) &= \frac{1}{\prod_{j=1}^m w_j^{p_j}} \\ \text{s.t. } \sum_{j=1}^m u_{ij} &= 1, i=1, 2, \dots, N \\ \sum_{i=1}^N u_{ij} &\geq 1, j=1, 2, \dots, m \\ u_{ij} &\in \{0, 1\}, \forall 1 \leq i \leq N, \forall 1 \leq j \leq m \\ p_j &= \frac{\sum_{i=1}^N u_{ij}}{N}, j=1, 2, \dots, m \end{aligned}$$

例如, 在特殊情况下, 当操作系统之间关联程度都为 0 时, 即全部的操作系统间不存在共同漏洞, 有

$$w_j = p_j, E(u) = \frac{1}{\prod_{j=1}^m p_j^{p_j}}$$

由条件 $\sum p_j = 1$ 易知, 当且仅当 $p_1 = p_2 = \dots = p_m = \frac{1}{m}$ 时有最大值, 最大值为 m 。证明如下。

证明 取 $f(u) = \ln \frac{1}{E(u)}$, 即 $f(u) = \sum_{j=1}^m p_j \ln p_j$,

设

$$f(u, \lambda) = \sum_{j=1}^m p_j \ln p_j + \lambda (\sum_{j=1}^m p_j - 1)$$

则有

$$f'_{p_j}(u, \lambda) = \ln p_j + 1 + \lambda, \forall 1 \leq j \leq m$$

令 $f'_{p_j}(u, \lambda) = 0$, 联立求解得 $p_1 = p_2 = \dots = p_m = e^{-(1+\lambda)}$, 又 $\sum p_j = 1$, 所以当且仅当 $p_1 = p_2 = \dots = p_m = \frac{1}{m}$, $f(u)$ 有极值存在, 对应 $E(u)$ 的最大值为 m 。此证明过程仅从理论角度进行分析, 实际情况要求 Np_j 的值为整数。

4.3 多样性操作系统配置算法

新用户申请虚拟机时, 只需要提供可以使用的操作系统类型以及所需虚拟机数量, CSP 就可以通过求解上述模型为用户推荐一个多样性最优的操作系统分配方案, 如算法 1 所示。如果是老用户申请, CSP 可以调用 hypervisor 的日志记录, 根据用户此前使用的操作系统记录, 以及当前的虚拟机状态, 计算出当前状态下的最优推荐策略。

算法 1 多样性操作系统配置算法

输入: N, m

输出: assignmentMap

- 1) **for each** $\{N, m\}$ from user in CSP **do**
- 2) **while** $m > 1 \& \& N > m$
- 3) **for each** $|OS_i| = n_1, |OS_j| = n_2, \dots, \sum n_n = N$
- 4) **get** $E(u)$ //对于不同的 OS 种类、数量组合, 计算多样性度量值
- 5) **find** $\{OS, m_m\} \leftarrow \max E(u)$ //利用上述模型找到令多样性最高的 OS 类型和数量
- 6) assignmentMap.add($VM_i \leftarrow OS_j$) //为每个 VM 配置 OS
- 7) **return** assignmentMap
- 8) **end**

4.4 理论分析

本文的威胁模型中, 认为攻击者发起一次完整的攻击过程后, 攻破虚拟机的概率为 1, 即可以对该虚拟机造成安全威胁, 如信息窃取或种植木马, 设定这一收益为固定值 b_1 , 且完整的攻击过程付出的成本代价为 c_1 。定义一个攻击效益 $e(s)$ 表示攻击者获得收益的总和与总成本的比值。例如, 攻击者对一个虚拟机进行完整攻击的攻击效益为

$$e(s_1) = \frac{b_1}{c_1} = \delta$$

其中, δ 为一个定值, s_1 为某种攻击策略。

攻击者由于受到成本的约束, 不可能对每一个虚拟机都进行完整攻击, 因此其选择的策略必

然是利用某次完整攻击获取的信息，且默认该服务器上用户虚拟机属于同一类型，从而利用完整攻击中挖掘的漏洞对其他虚拟机发起攻击。假设攻击者对操作系统 a 发起一次完整攻击后，对其他虚拟机利用相同漏洞继续攻击，那么显然，攻击操作系统 b 成功的概率为 x_{ab} 。另外，假设攻击者对单个虚拟机不进行预先探测和挖掘，直接利用共同漏洞进行攻击的成本为 c_2 ，不妨令 $c_1 = \alpha c_2$ ，由于完整攻击的成本高于直接攻击，所以有 $\alpha > 1$ 成立。对于攻击者，存在如下成本约束。

$$N_1 c_1 + (N - N_1) c_2 \leq C$$

其中， N_1 为完整攻击的次数，而 C 为总的成本。那么当操作系统不存在多样性时，攻击者只需要一次完整攻击，就可以攻破所有虚拟机，此时攻击效益

$$e(s_2) = \frac{N b_1}{c_1 + (N - 1) c_2}$$

当利用多样性操作系统，如 a 、 b 、 c 这 3 种异构的操作系统时，假设攻击者对使用操作系统 a 的虚拟机发起完整攻击，那么攻击效益为

$$e(s_3) = \frac{n_1 + n_2 x_{ab} + n_3 x_{ac}}{c_1 + (N - 1) c_2} b_1, \sum_{i=1}^3 n_i = N$$

其中， n_1, n_2, n_3 为 3 种操作系统的数量，由于 $0 < x_{ab}, x_{ac} < 1$ ，所以 $e(s_3) < e(s_2)$ 成立，即攻击效益降低。

5 虚拟机安全部署策略

对于 CSP 来说，来自同一用户虚拟机的部署方式是一个值得权衡的博弈问题，其原因如下。

1) 将实例集中于少数服务器的话，可以提高资源利用效率，然而一旦服务器需要维护升级，而此时某些关键服务又无法中断，那么同时在线迁移这些虚拟机需要保证足够的带宽和 CPU，产生的开销是巨大的；另外，集中部署的方式给攻击者提供了更广泛的攻击思路，如通过攻陷 hypervisor 直接获取所有虚拟机的权限，或者对服务器发起攻击，造成 DoS。

2) 将虚拟机分散到云环境中不同的主机上是一种负载均衡的有效方法，可以减轻虚拟机资源需求迅速增加时引起的性能降低问题，但是增加了用户的管理成本，大量工作状态的主机也会

消耗很多电力资源。因此，以不同的方式部署虚拟机，对于能耗、资源利用率和用户数据安全将产生不同的影响。

首先，本文认为将虚拟机分散到各个服务器中的策略降低了系统的安全性。由于 CSP 无法预先判断攻击者的存在，即攻击者有可能已经存在于云环境中。文献[18]提到，对于同一用户而言，个人拥有的虚拟机实例如果分散在不同的物理主机上，实际上增大了攻击者的攻击面，因为虚拟机有更大的概率暴露在攻击者的攻击环境下。因此，本文倾向于集中虚拟机的策略。

其次，之前已经提到，从用户的角度来看，过度的集中一定程度上影响了系统的顽健性，即服务器的失效会影响该服务器上用户所有的虚拟机。对此，本文设定一个阈值 T_m ，服务器上属于同一用户的实例数量不能超过该阈值。而在本文的应用场景中，还将对服务器上使用同一操作系统的虚拟机数量进行限制。

当云环境中已经存在该用户 VM 时，根据上述 2 个原则可以部署新的虚拟机。而如果虚拟机都被挂起，但有服务器部署过这些虚拟机时，显然应该首先在这些服务器上部署新的虚拟机，这仍然是从减小攻击面的角度出发。算法 2 为所提的安全部署算法。

算法 2 基于操作系统多样性的虚拟机安全部署策略算法

- 1) 对云环境中处于工作状态的服务器进行统计，集合为 $W = \{s_1, s_2 \dots\}$ ；
- 2) 为用户 $User_k$ 设置一个优先部署列表 $FPList_k = \{\}$ 和一个次优部署列表 $SPList_k = \{\}$ ；
- 3) **for each** server $s_i \in W$
- 4) **if** (s_i 上正在运行 $User_k$ 的虚拟机或曾经部署过该用户的虚拟机)
- 5) **&&** (s_i 上属于 $User_k$ 且处于启动状态的虚拟机数量少于 T_m)
- 6) $FPList_k.add(s_i)$ //将服务器 s_i 增加到优先列表中
- 7) **else**
- 8) $SPList_k.add(s_i)$ //将服务器 s_i 增加到次优先列表中
- 9) **end if**
- 10) **end for**

11) 对 $FPList_k$ 中的服务器按照拥有 $User_k$ 虚拟机的数量降序进行排序, 集合为 S_d ;

12) **while** $VMs_r \neq \emptyset$ // $User_k$ 需要部署的虚拟机集合非空

13) **for** ($i = 1; VM_r = get(VMs_r, i); i++$) **do**
//对于任意请求

14) **if** ($S_d.isEmpty == false$) //优先列表非空

15) **for** ($j = 1; j \leq |S_d|$) **do**

16) $S_s = get(S_d, j)$; //尽量使得服务器上同一用户的虚拟机数量达到饱和

17) **if** ($S_s.rest_resource == true \ \&\& \sum S_s.OSType_sameToVM_r < T_o$) //备选服务器剩余 CPU、带宽等资源满足虚拟机 VM_r 的资源需求, 且与 VM_r 的操作系统相同的虚拟机数量不超过阈值 T_o

18) $VM_r \rightarrow S_s$; //虚拟机 VM_r 部署到服务器 S_s 上

19) **else** $j++$; //不满足资源需求, 则考虑下一个服务器作为备选

20) **end if**

21) **end for**

22) **else** select $S_{rd} \in SPList_k, S_{rd}.rest_resource == true$ //优先列表中没有可选服务器时, 在次优列表中选择剩余资源充足的服务器

23) $VM_r \rightarrow S_{rd}$;

24) update physical resources; //更新物理资源信息

25) update W and S_d ;

26) **end if**

27) **end for**

28) **end while**

6 仿真结果

6.1 实验设置

首先, 本文利用 CVE (common vulnerabilities & exposures) 网站提供的数据库, 统计了一些常用操作系统的漏洞数及其与其他系统的共同漏洞数。需要指出的是, 考虑到不同操作系统的发行时间和发行商存在差异, 只统计了 2015 年~2017 年期间发现的漏洞数量, 如表 1 所示, 即由漏洞数得到的相关性矩阵 X 并不具有唯一性。

显然, 共同漏洞数对应的矩阵是对称矩阵, 但值得注意的是, 其对应的相关性矩阵并非对称矩阵。

本文首先针对用户申请虚拟机时, 如何配置操作系统以得到最大的多样性度量值进行实验, 分析操作系统种类、数量、相关性等以及虚拟机数量对多样性的影响; 然后在不同的多样性度量下, 比较攻击者在进行一次完整攻击时, 攻击效益的变化情况, 并与单一的操作配置方法进行对比; 最后研究在本文的安全部署策略下, 攻击者发起多次完整攻击, 探测到不同操作系统的概率与随机策略下这一概率的关系。

6.2 仿真结果分析

实验一 在多样性模型中, 本文证明了当操作系统之间的关联度为 0 时, 多样性度量值如何取得最大值。表 1 中有多种操作系统组合之间的共同漏洞数为 0, 因此实验 1 首先研究 3 种操作系统, 即 $m=3$ 时的分配方法。取 $N=10, m=3$ 进行仿真实验, 改变 3 种操作系统的选择, 分别计算使多样性度量指标最大时的分配方法, 具体结果如表 2 中的实验 1~实验 4 所示。

表 1 常用操作系统的共同漏洞数

OS	OS 共同漏洞					
	Windows 7	Windows 8	Windows 10	Ubuntu	Fedora	OS X
Windows 7	374	142	219	0	0	0
Windows 8	142	154	55	0	0	0
Windows 10	219	55	316	0	0	0
Ubuntu	0	0	0	583	58	31
Fedora	0	0	0	58	306	2
OS X	0	0	0	31	2	841

表 2 虚拟机数量一定，使用不同操作系统组合时，可以得到的多样性度量最大值

实验	操作系统						多样性度量
	Windows 7	Windows 8	Windows 10	Ubuntu	Fedora	OS X	
1	1	5	4	—	—	—	1.573 1
2	—	—	—	3	3	4	2.653 4
3	4	—	—	—	3	3	2.963 1
4	—	4	—	—	3	3	2.963 1
5	1	3	2	4	—	—	2.546 2
6	3	—	—	2	2	3	3.629 5
7	2	2	—	3	—	3	3.141 5
8	2	2	—	—	3	3	3.217 0

实验 1 中漏洞数量最多的 Windows 7 分配的虚拟机最少，实验 2 中 OS X 却分配到最多。因此，多样性的分配策略与操作系统具体漏洞数量无关，而是更容易受到与其他操作系统相关性的影响。

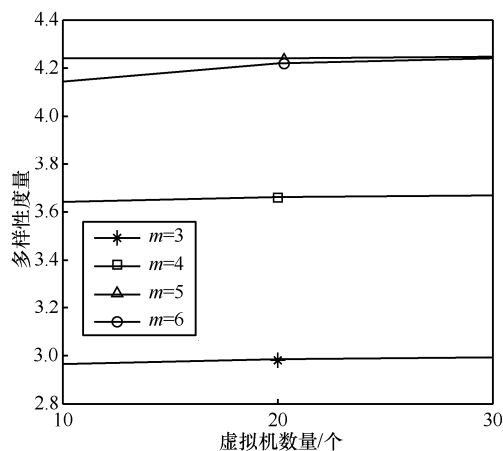
从实验 3、实验 4 发现，由于 Windows 操作系统与其他系统的相关性都为 0，所以当其他操作系统确定后，选择任何一种 Windows 系统不会对多样性产生影响。此时，CSP 可以结合用户的个人喜好，为用户推荐既保证安全性又满足用户需求的分配策略。

实验二 考虑增加一种操作系统类型时，多样性度量值是否会发生变化。即取 $N=10$ ， $m=4$ ，仿真结果如表 2 中实验 5~实验 8 所示。

从实验 1、实验 5 以及实验 2、实验 6 看出，增加分配操作系统的种类，可以明显提高虚拟机整体的多样性，但是需要指出，并不是操作系统种类越多，多样性程度就越高，如实验 5 与实验 2~实验 4 的对比，尽管增加了一种操作系统，但多样性却不如其他情况。因此，盲目地配置多种操作系统，并不能实现最大的安全性需求。

图 1 展示了使用 m 种操作系统，配置 N 个虚拟机时，多样性度量值的最大值随虚拟机数量的变化情况，其中 m 为表 1 中常用操作系统中的任意 m 种。容易发现，使用相同种类数量的操作系统时，多样性随着虚拟机数量增加呈现小幅度的增长；且 m 小于 6 时，配置相同数量的虚拟机， m 越多样性程度越高。另外， $m=6$ 时，其多样性度量略低于 $m=5$ 时的结果，这个现象也符合上

面提到的结论，即随意增加操作系统的种类，并不一定会对多样性产生积极的影响，只有当增加的操作系统与其他操作系统共同漏洞数少的情况下，增加 m 的大小，可以提高系统的安全性。

图 1 m 不同时，多样性度量值的最大值随虚拟机数量的变化情况

当然，计算多样性度量最大的操作系统配置方法，是以牺牲时间复杂度为代价的。例如， $m=3$ 时，需要计算的次数为 $C_6^3 \frac{(N-1)(N-2)}{2}$ ，随着虚拟机数量的增多，需要的计算时间将迅速增加。未来的工作中，可以使用模拟退火算法等优化算法来降低计算多样性的时间复杂度。

实验三 假设攻击者只发起一次完整攻击，分析其攻击效益在不同的多样性度量值下的变化情况。取 $N=10, \delta=0.5, \alpha=5$ （事实上， δ 和 α 的值对实验结果不产生影响，此处为了定量分析的方便），对应表 2 中的操作系统分配策略，分别计

算相应多样性度量值下的攻击效益。由于完整攻击只能攻破一种操作系统，因此实验计算出攻击任一操作系统时的攻击效益，并以最大值和最小值作为实验数据进行分析。实验结果如图2所示。

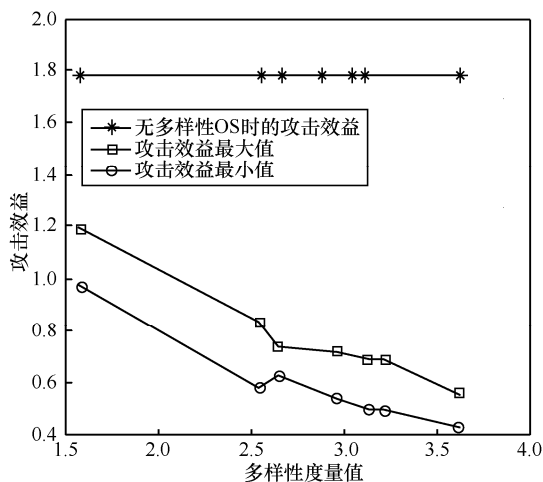


图2 攻击者发起一次完整攻击时，攻击效益与多样性度量值的关系

整体上，攻击效益会随着多样性度量值的增大而减小，分配策略不同时攻击效益可能出现相等的情况。多样性操作系统的策略相比于单一的操作系统分配，至少可以降低33.46%的攻击效益，且随着多样性度量值增加，降低效果更明显。

最后，本文对虚拟机部署后，攻击者针对不同的物理主机发起2次完整攻击（假设攻击者在成本约束下最多只能发起2次完整攻击）的情形做了简单分析。由于从完整攻击获取的信息对于直接攻击至关重要，因此该实验考虑攻击者在2次完整攻击中，是否能探测到2种不同的操作系统。取 $N=10$ ， $m=3$ 的情况，比较本文部署策略与随机策略下，探测到不同操作系统的概率，结果如图3所示，横轴括号中数字分别代表3种操作系统的数量。

可以发现，在本文部署策略下，攻击者探测到2种不同操作系统的概率略小于随机策略下的概率。随机策略是一种比较安全的部署方法，但由于随机策略并不考虑云环境中服务器的负载状态，容易造成服务水平协议（SLA, service level agreements）隔离，也可能使休眠状态的服务器重新启动，策略的可行性差。而本文所提部署策略可以实现与随机策略相当的安全性需求，同时，提高了服务器的资源利用率。

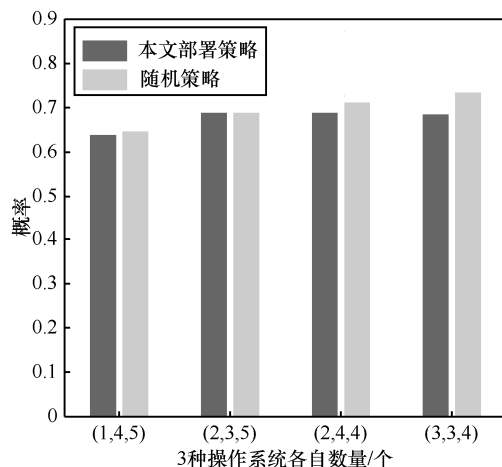


图3 攻击者发起2次完整攻击时，2种不同操作系统的概率与部署情况的关系

7 结束语

本文提出一种基于操作系统多样性的虚拟机安全部署策略。该方法考虑不同操作系统存在的漏洞数量和种类具有差异性，对虚拟机利用多样性的配置方法可以使攻击者无法以较小的开销攻陷所有的虚拟机。同时，考虑虚拟机在部署过程中需要实现资源节约等限制条件，本文提出一种安全部署策略，既能发挥多样性的优势，又不影响云环境的资源利用效率。实验结果表明，相比单一的操作系统的配置方法，本文方法通过提升系统多样性达到安全部署的目的，对于攻击者的攻击效益至少可以降低33.46%。

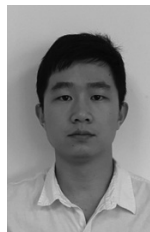
参考文献:

- [1] RISTENPART T, TROMER E, SHACHAM H, et al. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds[C]//ACM Conference on Computer and Communications Security. 2009.
- [2] APECECHEA G I, INCI M S, EISENBARTH T, et al. Fine grain cross-VM attacks on Xen and VMware are possible![J]. Ecewp.ece.wpi.edu, 2014.
- [3] OSVIK D A, SHAMIR A, TROMER E. Cache attacks and countermeasures: the case of AES[J]. Lecture Notes in Computer Science, 2006, (2005):1-20.
- [4] YAROM Y, FALKNER K. FLUSH+RELOAD: a high resolution, low noise, L3 cache side-channel attack[C]//Usenix Conference on Security Symposium. 2014:719-732.
- [5] ZHANG Y, REITER M K. Düppel: retrofitting commodity operating systems to mitigate cache side channels in the cloud[C]//ACM Sig-sac Conference on Computer & Communications Security. 2013.
- [6] WANG Z, LEE R B. A novel cache architecture with enhanced performance and security[C]//IEEE/ACM International Symposium

- on Microarchitecture. 2008.
- [7] MOON S J, SEKAR V, REITER M K. Nomad: mitigating arbitrary cloud side channels via provider-assisted migration[C]//The 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015:1595-1606.
- [8] AVIZIENIS A, CHEN L. On the implementation of N -version programming for software fault tolerance during program execution[J]. Proc of the Compsac, 1977, 6(1):25-37.
- [9] FORREST S, SOMAYAJI A, ACKLEY D. Building diverse computer systems[C]//The Workshop on Hot Topics in Operating Systems. 1997: 67-72.
- [10] HOFMEYR S A, FORREST S. Architecture for an artificial immune system[J]. Evolutionary Computation, 2000, 8(4):443-473.
- [11] GASHI I, POPOV P, STRIGINI L. Fault tolerance via diversity for off-the-shelf products: a study with SQL database servers[J]. IEEE Transactions on Dependable and Secure Computing, 2007, 4(4): 280-294.
- [12] NEWELL A, OBENSHAIN D, TANTILLO T, et al. Increasing network resiliency by optimally assigning diverse variants to routing nodes[C]//IEEE/IFIP International Conference on Dependable Systems and Networks. 2013.
- [13] GARCIA M, BESSANI A, GASHI I, et al. Analysis of operating system diversity for intrusion tolerance[J]. Software-practice & Experience, 2014, 44(6): 735-770.
- [14] BATES A, MOOD B, PLETCHER J, et al. Detecting co-residency with active traffic analysis techniques[C]//The 2012 ACM Workshop on Cloud Computing Security Workshop. 2012
- [15] ZHANG Y, JUELS A, OPREA A, et al. HomeAlone: co-residency detection in the cloud via side-channel analysis[C]//IEEE Symposium on Security and Privacy. 2011.
- [16] ZHANG Y, JUELS A, REITER M K, et al. Cross-VM side channels and their use to extract private keys[C]//ACM Conference on Computer and Communications Security. 2012
- [17] LEINSTER T, COBBOLD C A. Measuring diversity: the importance of species similarity[J]. Ecology, 2012, 93(3):477.
- [18] HAN Y, CHAN J, ALPCAN T, et al. Using virtual machine allocation policies to defend against co-resident attacks in cloud

computing[J]. IEEE Transactions on Dependable & Secure Computing, 2017, 14(1): 95-108.

作者简介:



张淼 (1994-), 男, 湖北孝感人, 国家数字交换系统工程技术研究中心硕士生, 主要研究方向为网络安全防御。

季新生 (1968-), 男, 江苏南通人, 国家数字交换系统工程技术中心教授、博士生导师, 主要研究方向为网络空间安全、拟态安全。

艾健健 (1989-), 男, 山东广饶人, 国家数字交换系统工程技术研究中心博士生, 主要研究方向为网络安全、软件定义网络。

刘文彦 (1986-), 男, 河南鹿邑人, 博士, 国家数字交换系统工程技术研究中心助理研究员, 主要研究方向为网络空间防御和云安全。

扈红超 (1982-), 男, 河南商丘人, 博士, 国家数字交换系统工程技术研究中心副研究员, 主要研究方向为网络空间安全、云数据中心。

霍树民 (1985-), 山西长治人, 博士, 国家数字交换系统工程技术研究中心助理研究员, 主要研究方向为网络空间安全。