基于动态概率攻击图的云环境攻击场景构建方法

王文娟，杜学绘，单棣斌
（信息工程大学，河南 郑州 450001）

摘 要：针对复杂多步攻击检测问题，研究面向云计算环境的攻击场景构建方法。首先，构建了动态概率攻击图模型，设计了概率攻击图更新算法，使之能够随着时空的推移而周期性更新，从而适应弹性、动态性的云计算环境。其次，设计了攻击意图推断算法和最大概率攻击路径推断算法，解决了误报、漏报导致的攻击场景错误、断裂等不确定性问题，保证了攻击场景的准确性。同时将攻击场景随动态概率攻击图动态演化，保证了攻击场景的完备性和新鲜性。实验结果表明，所提方法能够适应弹性、动态的云计算环境，还原出攻击者完整的攻击渗透过程，重构出高层次的攻击场景，为构建可监管可追责的云环境提供了一定的依据和参考。

关键词：云计算；攻击场景；动态概率攻击图；攻击意图；最大概率攻击路径
中图分类号：TP309.5
文献标识码：A
DOI: 10.11959/j.issn.1000−436x.2021004

Construction method of attack scenario in cloud environment based on dynamic probabilistic attack graph

WANG Wenjuan, DU Xuehui, SHAN Dibin
Information Engineering University, Zhengzhou 450001, China

Abstract: Aiming at the problem of complex multi-step attack detection, the method of attack scenario construction oriented to cloud computing environment was studied. Firstly, a dynamic probabilistic attack graph model was constructed, and a probabilistic attack graph updating algorithm was designed to make it update periodically with the passage of time and space, so as to adapt to the elastic and dynamic cloud computing environment. Secondly, an attack intention inference algorithm and a maximum probability attack path inference algorithm were designed to solve the uncertain problems such as error and fracture of attack scenarios caused by false positive or false negative, and ensure the accuracy of attack scenario. Meanwhile, the attack scenario was dynamically evolved along with the dynamic probability attack graph to ensure the completeness and freshness of the attack scenario. Experimental results show that the proposed method can adapt to the elastic and dynamic cloud environment, restore the penetration process of attacker's and reconstruct high-level attack scenario, and so provide certain references for building supervised and accountable cloud environment.

Keywords: cloud computing, attack scenario, dynamic probabilistic attack graph, attack intention, maximum probability attack path

1 引言

随着云计算[1]的发展和广泛应用，攻击者不再满足于传统的计算平台，而是将其主战场转移到了云计算平台。2016 年，云安全联盟（CSA, Cloud Security Alliance）公布了《云计算面临的十二大安全威胁》，其中高级可持续攻击（APT, advanced persistent threat）、分布式拒绝服务（DDoS, distributed

收稿日期：2020−07−23；修回日期：2020−09−23
基金项目：国家自然科学基金资助项目（No.61802436）；国家重点研发计划基金资助项目（No.2016YFB050190104）
Foundation Items: The National Natural Science Foundation of China (No.61802436), The National Key Research and Development Program of China (No.2016YFB050190104)

denial of service）攻击等仍将盛行。如图 1 所示的 DDoS 攻击过程中，攻击者给云环境中的虚拟机（VM, virtual machine）发送带有恶意代码 shellcode 的网址或文件，若 VM 使用带有漏洞的浏览器或应用程序打开恶意网址或恶意文件，shellcode 就会开始执行并利用浏览器或应用程序的漏洞获得该 VM 的控制权。攻击者下载恶意软件如木马到 VM 并安装木马软件，然后向木马软件建立远程连接下达攻击命令，例如控制木马软件扫描其他 VM 的漏洞、窃取被攻击 VM 里的重要信息、控制多个 VM 发动 DDoS 攻击。

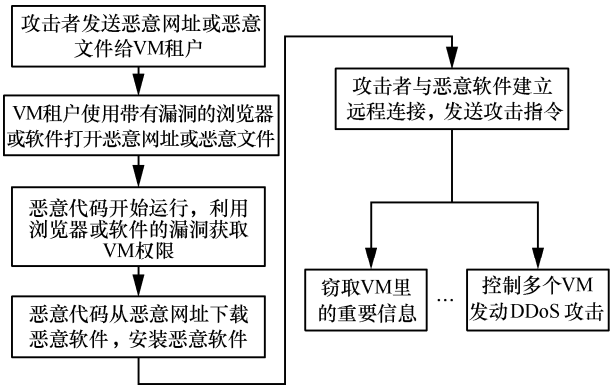

图 1 DDoS 攻击过程

可见，DDoS 攻击表现出复杂多步、隐蔽性强等特点，成为攻击者的重要攻击手段。复杂多步攻击是指一个完整的攻击由多个具有不同目的的单步攻击组成，达到某个目的之后继续实施下一个单步攻击，执行多个攻击步骤后实现最终目标。然而，云入侵检测系统（CIDS, cloud intrusion detection system）检测出的报警事件是孤立的、低层次的，反映的是基于单点的、单时间的单步攻击或攻击片段，往往不具备关联能力来识别复杂多步攻击。因此，研究如何揭示报警之间隐藏的关联关系、还原完整的攻击场景（AS, attack scenario）是应对复杂多步攻击的有效措施。攻击场景是指呈现攻击者入侵渗透过程的“画面”，通过“画面”能够获得攻击者发动攻击的起点、单步攻击之间的关系，以及攻击者的攻击目标、攻击路径等信息，攻击场景常用“图”的形式表示[2]。基于此，本文研究面向云计算环境的攻击场景构建方法，旨在再现和还原攻击者的复杂多步渗透过程，帮助云安全管理员从整体上把握攻击活动全貌，为实施有效的安全防御提供更全面的估计和决策依据。

2 相关工作

攻击场景构建实质上是报警关联分析，揭露各个孤立报警之间的逻辑关系，并转化为易于理解的、可视化的报警关联图。目前，报警关联所采用的技术手段主要包括以下 4 种：基于相似度的方法、基于数据挖掘的方法、基于属性攻击图的方法和基于概率攻击图（PAG, probability attack graph）的方法。

基于相似度的方法是通过计算报警之间各属性的相似度来判定报警之间是否具有关联关系，此种方法的关键在于定义相似度函数。Wang 等[3]通过计算报警间各属性相似度的加权平均和得到整体相似度，对整体相似度大于给定阈值的超报警进行关联。梅海彬等[4]提出利用报警间的相似度函数对具有相似攻击行为的序列进行聚类，并基于聚类的报警序列自动发现多步攻击模式。此种方法计算开销小，依赖专家知识较少，但需要预先定义阈值，只能发现统计上的关联情况。

基于数据挖掘的方法是通过关联算法、序列模式算法等发现隐藏在报警中的关联关系，依据关联关系构建攻击行为序列。葛琳等[5]利用 Map-Reduce 架构下的 Apriori 关联算法挖掘多维通信信息中的频繁项集，再进行综合关联分析。鲁显光等[6]使用改进的 FP Growth 算法挖掘报警之间的关联规则，继而进行报警关联。该方法不依赖专家知识，能够发现未知威胁行为，但是存在准确度较低、实用性较低等缺陷。

基于属性攻击图的方法直观地呈现了攻击者利用目标网络各脆弱点进行逐步渗透的所有可能攻击路径，描述了攻击步骤间的因果依赖关系，依据攻击图将报警进行匹配关联，从而构建攻击场景。因此，如何高效地生成攻击图和依据攻击图进行告警匹配关联成为该方法的 2 个重要研究内容。目前，一些研究更加专注于攻击图的分布式并行计算，以提高其生成效率。Wang 等[7]提出了一种基于启发式搜索的攻击图生成方法，通过引入匹配索引表来存储原子攻击的最新匹配结果，从而提高攻击图的生成效率。Kaynar 等[8]提出了一种分布式攻击图生成算法，应用于分布式多代理平台，由多个代理节点分别生成子攻击图，再由中心节点将子攻击图合并成全局攻击图。实验表明，分布式并行计算可以提高攻击图生成速度。

然而，实际上攻击步骤间的因果关系存在着不

确定性因素，例如，无法确定某攻击是否一定会成功，前一攻击步骤能否导致下一攻击步骤的发生等[9]。研究者将攻击场景构建视为不确定性推理问题，试图从概率推理的角度对攻击步骤之间的因果关系进行分析。刘威歆等[10]提出了基于攻击图的多源报警关联分析算法，能够综合应用图关系和阈值进行报警的联动推断和预测，从而构建攻击场景。吕慧颖等[9]深入分析了网络对抗在空间上的复杂性和时间上的动态性，采用状态转移图描述了威胁事件的时空关联。该方法存在状态空间爆炸问题，难以适应大规模网络，且只能发现一定时间内的报警关联。陈小军等[11]提出了一个面向内部攻击意图推断的概率攻击图模型，并设计了攻击意图推断算法以及最大概率攻击路径推断算法，但是这些算法并没有考虑误报警和漏报警产生的影响。王硕等[12]提出了基于因果知识网络的攻击场景构建方法，将因果知识网络分为因果关系和因果知识，因果关系依据专家知识定义，因果知识则是从报警中挖掘计算得出。该方法依赖专家知识，从具有较高误报率和漏报率的报警中挖掘的因果知识是不太准确的。为了直观地描述各类方法的特性，表 1 对不同报警关联方法进行了分析比较。

表 1　　不同报警关联方法对比

方法	优点	缺点
基于相似度的方法	计算开销小，依赖专家知识少	需定义阈值，只能发现统计上的关联
基于数据挖掘的方法	能够发现未知攻击	准确度较低、实用性较弱
基于属性攻击图的方法	直观展示攻击步骤间因果关系	没有考虑攻击步骤间的不确定关系，静态攻击图
基于概率攻击图的方法	攻击图和不确定性相结合，定性与定量分析，准确度较高	静态攻击图

综上所述，基于概率攻击图的方法能够揭示攻击步骤间的因果依赖和概率推理关系，准确度较高，此方法已成为目前攻击场景构建的主流方法。然而，基于概率攻击图的方法仍然面临一些问题和挑战。

1) 云计算具有弹性、动态性的特点，其网络拓扑结构、安全策略如防火墙规则等会经常发生改变，这导致概率攻击图不可能是一成不变的，而应是随着时间的推移动态变化的。

2) 异常检测系统不可避免地存在误报和漏报问题，误报和漏报无疑会对攻击场景的构建产生影响，如出现冗余、断裂等问题，并不能真正地反映攻击者的实际攻击过程。

针对以上问题，本文提出了一种基于动态概率攻击图的云环境攻击场景构建方法。该方法在概率攻击图的基础上，立足于攻击步骤间的因果依赖关系和概率推理关系，考虑其时空约束和动态特性，对动态概率攻击图模型、攻击场景构建过程进行了形式化建模和设计实现。本文的主要贡献如下。

1) 构建了动态概率攻击图模型，通过定性分析与定量分析相结合，能够准确合理地刻画攻击步骤间的因果依赖关系和概率推理关系。设计了概率攻击图更新算法，使之能够随时间的推移而动态变化，从而适应弹性、动态的云计算环境。

2) 在已知动态概率攻击图和观测报警的前提下，设计了攻击意图推断算法和最大概率攻击路径推断算法，解决了误报、漏报导致的攻击路径错误、断裂等不确定性问题，保证了攻击场景的准确性。同时，将攻击场景随动态概率攻击图动态演化，保证了攻击场景的完备性和新鲜性。

3) 在部署的云计算实验环境下对提出的动态概率攻击图模型以及攻击场景构建方法进行了实验验证。实验结果表明，本文方法能够构建完整的、动态演化的攻击场景，进而帮助云安全管理员从整体上把握云环境的安全态势。

3　动态概率攻击图构建

动态概率攻击图借鉴动态图思想，动态图又称图流或图序列，指会随时间推移而动态更新的图[13]。这里，为了方便理解和描述动态概率攻击图，首先定义概率攻击图。

3.1　动态概率攻击图定义

定义 1　概率攻击图。PAG 由网络结构和网络参数两部分组成，记为 PAG=(G, P)。

1) 网络结构 G=(V, E)

G 是一个有向无环图。

$V = A \cup S$ 为节点集合。A=$\{a_i | a_i$=action(host, vul)$\}$为动作节点集合，action 表示原子攻击 a_i 所采取的动作，可能是一次扫描或一次漏洞利用；host 表示 a_i 的攻击目标主机；vul 表示 a_i 利用的漏洞信息；变量 a_i 取值为 1 或 0，表示原子攻击行为是否发生。S=$\{s_i | s_i$=state(host)$\}$为状态节点集合，state 表示攻击者所处的状态；变量 s_i 取值为 1 或 0，表示

是否获取 host 的权限或占有 host 的资源，s_0 表示初始状态，s_i 表示攻击致变状态。$G=\{g_i|g_i=\text{goal(host)}\}$ 为攻击目标状态节点集合，满足 $G \subset S$。

$E=\{e_i \mid e_i \in (E_s \cup E_a)\}$ 为有向边集合。其中，$E_s = S \times A$ 为前提边，表示只有前置状态满足时才能实施某原子攻击；$E_a = A \times S$ 为后果边，表示原子攻击成功实施后能够达到某个新的后置状态。

2) 网络参数 $P=(\Delta,\Theta)$

$\Delta=\Delta_s \cup \Delta_a$ 表示依附在有向边上的权值。其中，Δ_s 依附于 E_s，指状态节点到动作节点的一步转移概率，表示在前置状态满足的条件下攻击发生的概率，称为攻击发生概率；Δ_a 依附于 E_a，表示原子攻击成功后到达后置状态的概率，称为攻击成功概率。

$\Theta=\{\theta_i \mid \theta_i = P(v_i \mid \text{Pre}(v_i))\}$ 表示局部条件概率分布表（LCPT, local conditional probability table）。θ_i 表示节点 v_i 与其所有父节点的依赖关系程度，Pre(v_i)表示节点 v_i 的父节点集合。根节点 s_0 没有父节点，其概率值由先验知识给出，根据根节点的概率值和边权值便可推导出所有后续节点的 LCPT。

定义 2 概率攻击图更新操作。概率攻击图更新操作可以用三元组<op, a_i, s_j>表示。其中，op={I, D}表示更新操作类型，op=I 表示增加操作，op=D 表示删除操作，a_i 和 s_j 分别表示概率攻击图中与操作 op 相关的动作节点和状态节点。这里用更新操作集合 $\text{GC}_i=\{<\text{op}_1,a_1,s_1>,\cdots,<\text{op}_k,a_k,s_k>\}$ 表示在 t_i 时刻的所有更新操作。

需要说明的是，向概率攻击图中增加一个节点，包括与该节点相关的一系列增加边、增加边权值和 LCPT 等操作。类似地，删除一个节点，则包括与该失效节点相关的一系列删除边、删除边权值等操作。

为了描述概率攻击图动态变化的特性，需要引入时间因素，这里将攻击步时作为概率攻击图的更新周期。

定义 3 攻击步时 Δt。当攻击者的入侵行为有一个攻击步时，若在长时间内仍未发起后续攻击，则认为入侵失败，设置一个时间窗口 Δt 来衡量攻击的成功与否。已知大部分攻击的攻击步时为 2 h，设置 $\Delta t = 2\text{ h}$。

定义 4 动态概率攻击图（DPAG, dynamic probability attack graph）。DPAG 指在时间域 $T=[t_0, t_n]$内随攻击步时 Δt 动态更新的概率攻击图流，可表示为

$$\text{DPAG}^T = \{<\text{PAG}_i, t_i> \mid 0 \leqslant i \leqslant n, t_i = t_{i-1} + \Delta t\}$$

$$\text{GC}_i : \text{PAG}_{i-1} \mid\!\rightarrow \text{PAG}_i,\ \ \text{GC}_0 = \varnothing$$

其中，二元组<PAG_i, t_i>表示 PAG_0 在 t_i 时刻上的更新图 PAG_i，PAG_0 表示 t_0 时刻上的初始概率攻击图。GC_i 表示 t_i 时刻的更新操作，$\text{GC}_i : \text{PAG}_{i-1} \mid\!\rightarrow \text{PAG}_i$ 表示 t_{i-1} 时刻的概率攻击图 PAG_{i-1} 在更新操作 GC_i 下得到的 t_i 时刻的概率攻击图 PAG_i。

由定义 4 可知，动态概率攻击图的构建过程实际上是初始概率攻击图的构建与更新过程，其更新过程是按固定时间间隔的（即周期性更新），适应于较稳定的、改变量较小的私有云环境。对于公有云来说，其环境规模大，变化频繁，如新增大量节点、部署新的系统等，为了提高更新效率，可以采取触发更新机制，即当网络环境发生变化时，提前自适应地局部更新。

3.2 概率攻击图构建

概率攻击图的构建主要包括两部分：网络结构生成和网络参数确定。网络结构生成是指生成动作节点和状态节点，并识别节点之间的因果依赖关系，以有向无环图直观展示；网络参数确定是指确定节点间的因果依赖强度。

3.2.1 网络结构生成

网络结构生成是依据拓扑结构与漏洞信息等，借助攻击图生成工具生成攻击图的网络结构。目前，研究者已开发了多种攻击图生成工具，如 Ou 等[14]开发的多主机多漏洞分析工具 MulVAL、Noel 等[15]提出的拓扑脆弱性分析工具 TVA、Lippmann 等[16]提出的基于图论的攻击图生成工具 NetSPA 等。其中，MulVAL 是 Linux 平台开源攻击图生成工具，可用于大规模网络的攻击图生成，实验数据表明，对于包含 2 000 台主机的模拟网络，生成攻击图的时间在 15 s 左右；对于具有 n 个节点的网络，复杂度为 $O(n^2)$，符合多项式时间要求。MulVAL 工具将主机/网络配置、网络连通性信息、漏洞信息等输入文件 input.P 中，使用 Graphviz 图形生成器绘制攻击图，输出的攻击图存储在 AttackGraph.PDF 文件中。由于 MulVAL 是开源工具，且相对于其他工具有更好的准确度和可扩展性，得到了工业界和学术界的一致认可，已广泛应用于攻击图可行性验证和性能测评，因此，本文选用 MulVAL 工具生成

网络结构。

3.2.2 网络参数确定

网络参数确定是指确定有向边权值以及每个节点的 LCPT。

1) 有向边权值确定

攻击发生概率 Δ_s。元素 $\Delta_{S_{ij}}$ 是前提边 (s_i, a_j) 的权值，表示在状态 s_i 满足的前提下攻击 a_j 发生的概率，$\Delta_{S_{ij}} = P(a_j \mid s_i)$。

攻击发生的概率与漏洞被利用的难易程度相关，漏洞越难被利用，相应的攻击发生的可能性越小。通用脆弱性评估系统（CVSS, common vulnerability scoring system）是漏洞评估的公开行业标准[17]。CVSS 中的访问向量（AV, access vector）、访问复杂度（AC, access complexity）、认证（Au, authentication）等属性描述了漏洞被利用的难易程度[17]，可作为确定攻击发生概率 Δ_s 的依据。攻击发生的概率可定义为

$$\Delta_s = 2\mathrm{AV} \times \mathrm{AC} \times \mathrm{Au} \tag{1}$$

实际上，也存在一些攻击并不是通过漏洞利用来发动的，如口令猜测攻击等。这类攻击发生的概率可以根据其发生的频率来确定。本文确定的攻击发生概率的取值如表 2 所示。

表 2　攻击发生概率的取值

指标		等级评分（频率）	Δ_s
漏洞利用难度	AV	本地 L：0.359；邻近 A：0.646；远程 N：1.0	$2\mathrm{AV}\times \mathrm{AC}\times \mathrm{Au}$
	AC	高 H：0.35；中 M：0.61；低 L：0.71	
	Au	多次 M：0.45；单次 S：0.56；不需要 N：0.704	
发生次数		近一周<5 次	0.1
		近一周<20 次	0.5
		近一周>20 次	0.8

攻击成功概率 Δ_a。元素 $\Delta_{a_{ij}}$ 是后果边 (a_i, s_j) 的权值，表示原子攻击 a_i 成功发生后到达后置状态 s_j 的概率，$\Delta_{a_{ij}} = P(s_j \mid a_i)$。

实际上，对于漏洞利用攻击来说，其攻击成功的概率与攻击发生的概率是相关的，比如攻击者选择一个漏洞进行攻击是否成功与该漏洞被利用的难易程度相关。因此，这里将漏洞利用攻击的成功概率置为 1.0，攻击发生概率中已经考虑了其难度。但是对于其他攻击，其攻击成功的概率与攻击发生的概率又不完全一样，比如口令猜测攻击很容易发生，但是成功的可能性却比较低。因此，其他攻击的成功概率根据专家知识库来设置，如果攻击难度较大，则攻击成功的概率为 0.1。攻击成功概率的取值如表 3 所示。

表 3　攻击成功概率的取值

攻击类型		Δ_a
漏洞利用		1.0
其他	容易	0.9
	一般	0.5
	难	0.1

2) 节点 LCPT 确定

概率攻击图中包括动作节点和状态节点 2 类，故 LCPT 的计算存在以下 2 种情况。

①指向同一动作节点 a_i 的各状态节点之间是“与”关系，即只有当攻击 a_i 的所有前置状态都满足时，a_i 才有可能发生。

②指向同一状态节点 s_i 的各动作节点之间是“或”关系，即任何一个原子攻击成功都能够达到此状态 s_i。

对于动作节点 a_i，其父节点集用 $\mathrm{Pre}(a_i)$来表示，则 a_i 的 LCPT 为

$$P(a_i \mid \mathrm{Pre}(a_i)) = \begin{cases} 0, \exists s_j \in \mathrm{Pre}(a_i), s_j = 0 \\ \prod_{s_j=1} \Delta_{s_{ji}}, \text{其他} \end{cases} \tag{2}$$

对于状态节点 s_i，其父节点集用 $\mathrm{Pre}(s_i)$来表示，则 s_i 的 LCPT 为

$$P(s_i \mid \mathrm{Pre}(s_i)) = \begin{cases} 0, \forall a_j \in \mathrm{Pre}(s_i), a_j = 0 \\ 1 - \prod_{a_j=1} (1 - \Delta_{a_{ji}}), \text{其他} \end{cases} \tag{3}$$

概率攻击图模型如图 2 所示。其中，虚线圆形表示状态节点，虚线方形表示动作节点，有向边表示节点间的因果关系。设节点 s_0 的先验概率 $P(s_0)$=0.8，根据式(2)和式(3)，可推导出所有后续节点的 LCPT。

不难发现，概率攻击图中网络结构 G 从定性的角度直观地刻画了原子攻击间的因果依赖关系，网络参数 P 则从定量的角度量化了因果依赖的强度，通过定性分析与定量分析相结合，有利于准确合理地构建攻击图。

3.3 概率攻击图更新

概率攻击图更新包括 2 种方法：重新生成和局

部更新。如果抛开原攻击图重新生成攻击图，则计算代价较大，新攻击图相较于原攻击图大部分是重复的，为此可以在原攻击图的基础上实现攻击图的动态更新。

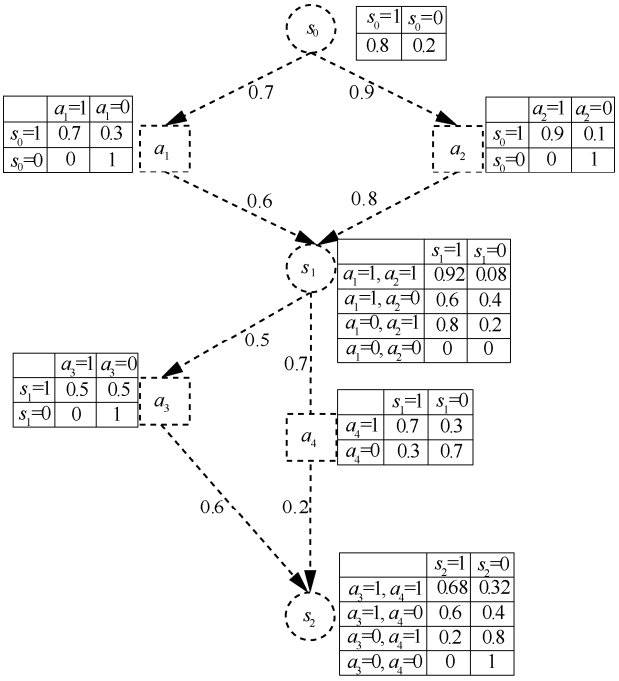

图 2 概率攻击图模型

引起概率攻击图变化的主要因素包括网络拓扑改变、主机节点信息改变、防火墙规则微调等，根据定义 2 可知，概率攻击图变化实际上是节点增加或删除、相关前提边和后果边增加或删除以及网络参数的改变。因此，概率攻击图的更新工作主要包括 3 个步骤：①确定引起改变的因素；②计算其改变量；③依据改变量更新概率攻击图。需要说明的是，当概率攻击图的网络结构发生变化时，网络参数也随之变化，这里为了简化描述，省略了网络参数的更新描述。下面分别从网络拓扑改变、主机节点信息改变、防火墙规则改变这 3 种情况来描述攻击图的局部更新过程。

1) 网络拓扑改变

当引起攻击图结构变化的因素是网络拓扑改变，即增加或删除一个主机节点时，需要确定该主机节点与其他节点的可达性关系，并计算可达性关系所产生的攻击图改变量。具体地，当增加一个主机节点时，解析该主机节点上的一个或多个脆弱性，在攻击图中增加与脆弱性相对应的动作节点，并基于可达性关系搜索动作节点的前置状态，使用前提边进行连接，若搜索不到，则从初始状态开始连接，同时增加新动作节点的后果边和后置状态；当删除一个主机节点时，解析该主机节点上的脆弱性，删除与脆弱性相对应的、失效的动作节点，并删除与失效动作节点相连接的所有边和后续节点。

2) 主机节点信息改变

当引起攻击图结构变化的因素是主机节点信息改变，即某主机的脆弱性信息改变、端口开放或关闭时，其更新过程与网络拓扑改变时相似。不同之处在于，当修复某主机的脆弱性时，在攻击图中应当删除失效的动作节点和后置状态节点。如果该动作节点的后置状态还存在其他父节点，则表明攻击者通过其他攻击行为也能够达到该状态，因此不删除其后置状态节点。

3) 防火墙规则改变

防火墙规则改变是指规则的增加和删除。当增加防火墙规则时，主机节点间的可达性关系可能会失效，由此攻击图内与规则相关的源主机节点和目的主机节点间的脆弱性关联关系也会失效；当删除防火墙规则时，主机节点间的可达性关系可能会增加，攻击图内与规则相关的主机节点间会产生新的脆弱性关联关系，其更新过程与网络拓扑改变时相似。

以网络拓扑改变为例，描述概率攻击图更新算法，如算法 1 所示。

算法 1 概率攻击图更新算法

输入 t_{i-1} 时刻的概率攻击图 PAG=(A, S, E)，新增的主机节点 host

输出 t_i 时刻的概率攻击图 PAG′=(A', S', E')

1) begin update algorithm
2) initialization PAG′=PAG
3) function Generate Topology (host)
4) {if (add host) //增加主机节点
5) generate VUL ∈ host //解析节点的漏洞
6) for each vul_i in VUL//增加对应动作节点
7) { add node $a_i \leftarrow \text{vul}_i$ to PAG′
8) if (Pre(a_i) exist) //搜索前置状态节点
9) add edge $\text{Pre}(a_i) \rightarrow a_i$ //增加前提边
10) else
11) add edge $s_0 \rightarrow a_i$

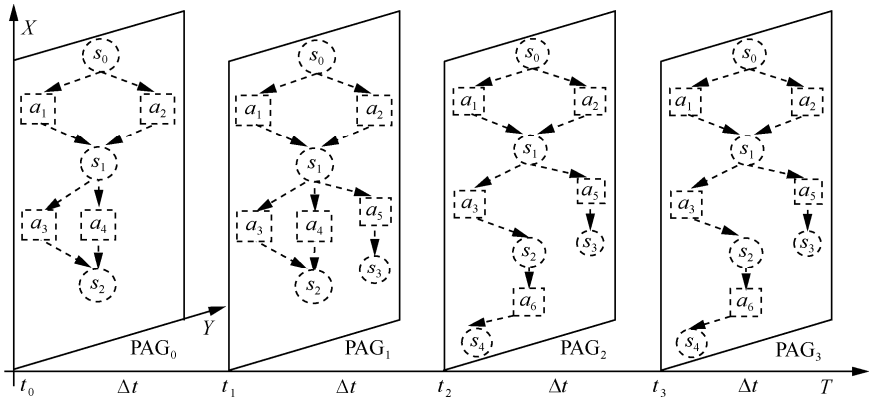

图 3　概率攻击图更新示例

12)　　add node $s_j = \mathrm{Post}(a_i)$ //增加后置状态节点

13)　　add edge $a_i \to s_j$ //增加后果边 }

14)　else (delete host)　　//删除主机节点

15)　generate VUL ∈ host

16)　for each vul_i in VUL //删除对应节点和边

17)　　{delete node $a_i \leftarrow \mathrm{vul}_i$ from PAG′

18)　　delete node $s_j = \mathrm{Post}(a_i)$

19)　　delete edge $\mathrm{Pre}(a_i) \to a_i$ //删除相关边

20)　　delete edge $a_i \to s_j$ } }

21) return PAG′

概率攻击图更新示例如图 3 所示。其中，X 轴和 Y 轴为空间方向，T 轴为时间方向。动态概率攻击图的时间域为 T=[t_0, t_3]，t_0 时刻对应的是初始概率攻击图 PAG_0=(A_0, S_0, E_0)。假设在 $t_1 = t_0 + \Delta t$ 时刻主机 host_1 中产生一个新的脆弱性，则攻击图中应该增加一个相对应的动作节点 a_5，同时增加节点 a_5 的前提边、后置状态 s_3 及后置边。用 PAG_1=(A_1, S_1, E_1)表示由 PAG_0 更新得到的概率攻击图，其更新操作可形式化描述为 $\mathrm{GC}_1 : \mathrm{PAG}_0 \mid\!\to \mathrm{PAG}_1$，$\mathrm{GC}_1 = \{< I, a_5, s_3 >\}$。类似地，假设在 t_2=t_1+Δt 时刻修复了主机 host_1 中的一个脆弱性，同时新增了一个主机 host_2 并产生一个新的脆弱性，则攻击图同时执行删除和增加更新操作。用 PAG_2=(A_2, S_2, E_2)表示 t_2 时刻更新得到的攻击图，更新操作可形式化描述为 $\mathrm{GC}_2 : \mathrm{PAG}_1 \mid\!\to \mathrm{PAG}_2$，$\mathrm{GC}_2 = \{< D, a_4 >, < I, a_6, s_4 >\}$。从图 3 中可以看出，动作节点 a_4 的后置状态 s_2 还存在其他父节点如 a_3，则表明攻击者通过原子攻击 a_3 也能够到达状态 s_2，因此不需要删除后置状态节点 s_2。在 t_3=t_2+Δt 时刻上的攻击图 PAG_3 相较 PAG_2 并无任何变化，因此更新操作可描述为 GC_3:$\mathrm{PAG}_2 \mid\!\to \mathrm{PAG}_3$，$\mathrm{GC}_3 = \varnothing$。

综上所述，概率攻击图更新操作分别是 $\mathrm{GC}_1 = \{< I, a_5, s_3 >\}$，$\mathrm{GC}_2 = \{< D, a_4 >, < I, a_6, s_4 >\}$，$\mathrm{GC}_3 = \varnothing$。时间域 T 内的动态概率攻击图序列为 $\mathrm{DAG}^T = \{< \mathrm{PAG}_0, t_0 >, < \mathrm{PAG}_1, t_1 >, < \mathrm{PAG}_2, t_2 >, < \mathrm{PAG}_3, t_3 >\}$。动态概率攻击图展示了攻击者从初始状态出发，利用目标网络的脆弱性发动各种可能的攻击动作，不断地从一个攻击状态转移到另一个新的攻击状态，最终达到攻击目标的过程。攻击者实施的攻击动作及经历的状态转移构成一个集合，称之为攻击路径或攻击链。

定义 5　攻击路径 path。攻击路径是指由概率攻击图中的状态节点和动作节点按照一定的因果依赖关系排列而成的攻击序列，记为 $\mathrm{path}(s_0, g) = \{s_0, a_1, s_i, a_j, \cdots, g\}$。其中，$\{s_0, s_i, \cdots, g\}$ 表示攻击状态转移序列，$\{a_1, a_j, \cdots, a_n\}$ 表示原子攻击行为序列。

概率攻击图是包含多条攻击路径的集合，即 $\mathrm{PAG} = \{\mathrm{path}^k(s_0, g)\}$。其中，$\mathrm{path}^k(s_0, g)$ 表示从初始状态 s_0 到终止状态 g 的编号为 k 的攻击路径。

4　攻击场景构建

报警来源于攻击且能够反映攻击，每一条 IDS 报警都反映了一个原子攻击的发生，利用概率攻击图作为模板，通过将报警与概率攻击图的动作节点进行映射，从而实现各孤立报警的关联，还原攻击者的攻击轨迹。依据动态概率攻击图，将攻击场景构建分为初建、重构和演化 3 步。

4.1　攻击场景初建

4.1.1　报警聚类

目标网络在同一时间段内可能受到来自多个

攻击者的入侵，这导致报警序列中包含了可能不止一个攻击者的入侵活动，如果不做处理而直接进行映射，则将导致攻击场景混乱。报警聚类旨在将隶属于同一攻击者的报警聚合到同一类簇中，以区分不同攻击者的入侵活动。

为了更好地说明，首先定义报警的相关概念。

定义 6 报警。报警是指原子攻击发生时被 IDS 观测到的事件，用 O 表示报警序列，$O=\{o_i|o_i=<\text{time, srcIP, srcPort, dstIP, dstPort, class}>\}$。报警 o_i 是一个六元组，其中，time 表示 IDS 产生报警的时间；srcIP 和 srcPort 分别表示报警的源 IP 地址和源端口；dstIP 和 dstPort 分别表示报警的目的 IP 地址和目的端口；class 表示报警的类型，表明具体发生了什么攻击。

IDS 产生的报警存在一些重复报警，主要是由同一攻击源用同一种攻击方式对目的主机进行多次不同时间的攻击所造成的，有必要消除重复报警，将其简化为同一条报警。此外，同一攻击活动触发的报警，彼此在 IP 地址分布上总是具有关联性，如前一攻击步骤的 dstIP 可能就是下一攻击步骤的 srcIP。依据这种 IP 地址相关性，将同一攻击活动的报警聚合在一起，从而避免关联关系混乱。

定义 7 重复报警。如果两条报警 o_i 和 o_j 的<srcIP, srcPort, dstIP, dstPort, class>字段都相等且在攻击步时 Δt 内（o_j.time−o_i.time< Δt ），则认为 o_i 和 o_j 具有重复关系，应该合并。

定义 8 IP 地址相关性。如果报警 o_i 的地址无论是源 IP 地址还是目的 IP 地址，总有一个和 o_j 的地址相同，且 o_i 和 o_j 在攻击步时 Δt 内，则认为 o_i 和 o_j 具有地址相关性。

定义 9 报警类簇。报警类簇是指将具有 IP 地址相关性的报警按照其产生的时间顺序排列而成的报警序列，记为 $C=\{o_1, o_2, \cdots, o_m\}$，满足 $C \subset O$。

基于IP地址相关性的报警聚类过程采用文献[18]所提方法，这里不再论述。若 2 个报警间是 IP 地址相关的，则划分到同一报警类簇中，从而将每个攻击者的报警区分开。

4.1.2 报警映射

报警映射是将报警 o_i 映射到概率攻击图中的节点，称为观测节点，并判断观测节点 o_i 的目的 IP 地址和类型是否与某动作节点 a_i 的主机和漏洞信息相匹配，若匹配则映射成功，否则映射失败为空集。定义映射函数如下。

定义 10 报警映射函数。

$$\text{map}(o_i, \text{PAG}) \rightarrow \begin{cases} a_i, \exists a_i \in \text{PAG} \wedge (o_i.\text{dstIP} = a_i.\text{host}) \wedge (o_i.\text{class} = a_i.\text{vul}) \\ \varnothing, \text{其他} \end{cases} \tag{4}$$

映射失败的原因可能有以下几种：①存在误报警导致攻击 a_i 没有发生，但却存在报警 o_i 与之匹配；②存在漏报警使攻击路径中的某动作节点 a_i 缺乏报警与之匹配；③概率攻击图的结构不完整。上述 3 种情况都会导致报警映射失败。具体地，报警映射情景如图 4 所示。图 4 中，浅灰色圆圈表示映射成功的观测节点，深灰色圆圈表示映射失败的观测节点，实线表示攻击实施成功或状态转移成功，虚线表示攻击实施失败或状态转移失败。

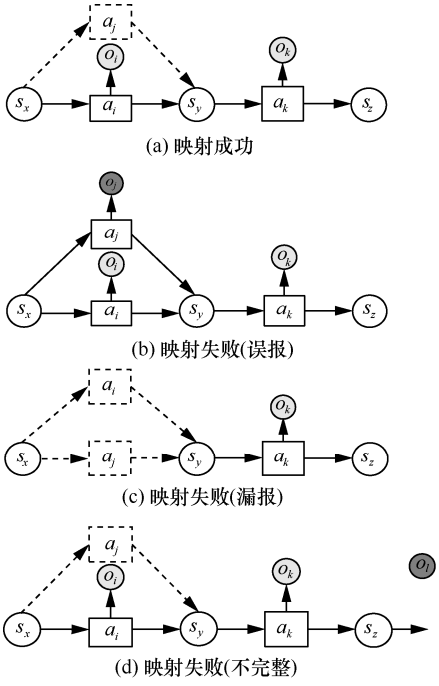

图 4 报警映射情景

图 4(a)表示报警映射成功。观测节点 o_i 和 o_k 分别与动作节点 a_i 和 a_k 相匹配，表明原子攻击 a_i 和 a_k 实施成功，攻击状态转移成功，状态 s_x、s_y 和 s_z 皆为 True。

图 4(b)表示误报引起的映射失败。通过报警映射得到攻击序列 $\{s_x, a_i, s_y, a_k, s_z\}$ 和 $\{s_x, a_j, s_y, a_k, s_z\}$，然而由于 o_j 是误报，攻击 a_j 实际上并未发生，因此映射得到的攻击路径 $\{s_x, a_j, s_y, a_k, s_z\}$ 是错误或冗余的。

图 4(c)表示漏报引起的映射失败。状态 s_x、s_y

和 s_z 为 True，表明存在原子攻击导致状态转移成功。然而由于漏报导致映射得到的攻击序列 $\{s_y,a_k,s_z\}$ 是不完整的、断裂的，从而无法确定攻击路径是 $\{s_x,a_i,s_y,a_k,s_z\}$ 还是 $\{s_x,a_j,s_y,a_k,s_z\}$。

图 4(d)表示概率攻击图结构不完整引起的映射失败。图中存在观测节点 o_l 但缺乏与之匹配的动作节点，故需更新攻击图。

可见，报警映射能够获得初步的攻击场景，但是由于误报、漏报问题，导致某些攻击片段存在冗余或者断裂，从而影响了对实际攻击目标和攻击路径的判断。因此，为了得到准确可信的攻击场景，需要对初步构建的攻击场景进行修正和重构，一方面通过衡量报警的置信水平，识别低置信度的、无效的报警证据，从而发现误报引起的冗余路径；另一方面计算各状态节点被入侵的概率，从候选攻击目标和攻击序列中推断最大可能的攻击目标和攻击序列。

4.2　攻击场景重构

攻击场景重构本质上是概率推理问题。在概率推理中，先验概率表示事件发生的概率，后验概率则表示当新的证据出现时变量发生的概率。那些值已经被确定的变量集合称为证据，而需要求解的变量集合称为假设，概率推理问题就是求解在给定证据变量 E 的条件下假设变量 H 的后验概率 $P(H\mid E)$。当不确定假设集合 H 中某一个假设 $h\in H$ 最佳时，最大后验估计认为具有最大后验概率（MAP, maximum a posteriori）的假设为最佳假设。即

$$h_{\mathrm{MAP}}=\arg\max_{h\in H}P(h\mid E) \tag{5}$$

基于此，攻击场景重构问题可形式化描述如下。

给定候选攻击序列集合 $H=\{h_i\mid h_i=\mathrm{path}^k(s_0,G)\}$ 和报警证据 $O=\{o_i\mid_{i=1}^m\}$，求解报警置信度 $\mathrm{PO}=\{\mathrm{Po}_i\mid_{i=1}^m\}$、攻击意图 $g_{\mathrm{MAP}}=\arg\max_{g\in G}P(G\mid O,\mathrm{PO})$ 和最大概率攻击路径 $h_{\mathrm{MAP}}=\arg\max_{h\in H}P(H\mid O,\mathrm{PO})$。

4.2.1　报警证据识别

定义 11　报警置信度 PO。PO 是指在观测到报警事件 o_i 的情况下，能证明其对应的原子攻击 a_i 发生的概率 $P(a_i\mid o_i)$，报警置信度也即动作节点 a_i 的后验概率。

设 IDS 对攻击 a_i 的检测率和误报率分别为 d_i 和 f_i，检测率是指在攻击发生的情况下，IDS 产生报警的概率；误报率是指在攻击没有发生的情况下，IDS 却产生报警的概率，故存在 $P(o_i\mid a_i)=d_i$，$P(o_i\mid\neg a_i)=f_i$，报警置信度 $P(a_i\mid o_i)$ 为

$$P(a_i\mid o_i)=\frac{P(o_i\mid a_i)P(a_i)}{P(o_i)}=\frac{P(o_i\mid a_i)P(a_i)}{P(o_i\mid a_i)P(a_i)+P(o_i\mid\neg a_i)P(\neg a_i)}=\frac{d_iP(a_i)}{d_iP(a_i)+f_iP(\neg a_i)} \tag{6}$$

其中，$P(a_i)$ 为动作节点 a_i 的先验概率，表示攻击 a_i 发生的概率；$P(\neg a_i)=1-P(a_i)$ 表示攻击 a_i 未发生的概率。

在概率攻击图中，某节点 v_i 的先验概率 $P(v_i)$ 可以通过下面 2 种方法求解。

方法 1　通过求解边缘概率分布得到，已知节点的 LCPT，便可获得节点的先验概率为

$$P(v_i)=\sum_{v_1,\cdots,v_{i-1},v_{i+1},\cdots,v_n}P(v_1,\cdots,v_i,\cdots,v_n)=\sum_{v_1,\cdots,v_{i-1},v_{i+1},\cdots,v_n}\left(\prod_{i=1}^{n}P\left(v_i\mid\mathrm{Pre}(v_i)\right)\right) \tag{7}$$

方法 2　通过父节点的先验概率以及节点间的转移概率求解得到。

动作节点 a_i 的先验概率 $P(a_i)$ 为

$$P(a_i)=\begin{cases}0,\exists s_j\in\mathrm{Pre}(a_i),s_j=0\\ \prod_{s_j=1}P(s_j)\Delta_{s_{ji}},\text{其他}\end{cases} \tag{8}$$

状态节点 s_i 的先验概率 $P(s_i)$ 为

$$P(s_i)=\begin{cases}0,\forall a_j\in\mathrm{Pre}(s_i),a_j=0\\ 1-\prod_{a_j=1}(1-P(a_j)\Delta_{a_{ji}}),\text{其他}\end{cases} \tag{9}$$

通过上述 2 种方法可以求得节点 a_i 的先验概率 $P(a_i)$，并进一步得到报警 o_i 的置信度 $P(a_i\mid o_i)$。

通过报警置信度能够衡量报警的置信水平，识别有效的报警证据。一方面，报警 o_i 的置信度 $P(a_i\mid o_i)$ 应当大于 50%，否则，$P(\neg a_i\mid o_i)$ 将大于 50%，这表明在证据 o_i 出现时，攻击未发生或攻击失败的概率大于 50%，故报警 o_i 为误报警；另一方面，攻击 a_i 的后验概率 $P(a_i\mid o_i)$ 应当大于其先验概率 $P(a_i)$，这表明在报警证据 o_i 出现时，攻击 a_i 发生的可能性提高了；反之，当后验概率 $P(a_i\mid o_i)$ 小于先验概率 $P(a_i)$ 时，则表明证据 o_i 的出现反而降

低了攻击 a_i 发生的可能性，这与实际情况不符，因此认为报警 o_i 为误报警。基于上述分析，可以得出以下结论。当且仅当 a_i 和 o_i 满足 $P(a_i \mid o_i) > 50\%$ 和 $P(a_i \mid o_i) > P(a_i)$ 这 2 个条件时，原子攻击所产生的报警为真实报警。

4.2.2 攻击意图推断

攻击意图推断是指已知目标状态节点集合 G，在给定报警证据 $O = \{o_i \mid_{i=1}^{m}\}$ 及其置信度 $\mathrm{PO} = \{\mathrm{Po}_i \mid_{i=1}^{m}\}$ 的情况下，计算并比较各目标状态节点被入侵的概率，从而获得具有最大后验概率 MAP 的目标状态节点，即 $g_{\mathrm{MAP}} = \arg\max_{g \in G} P(G \mid O, \mathrm{PO})$。

状态节点 s_i 与证据 O 并无直接的联系，然而证据 O 的出现会提高原子攻击发生的可能性，进而提高其后置状态存在的可能性。因此，状态节点 s_i 的后验概率与其父节点 $\mathrm{Pre}(s_i)$的后验概率有关。

以图 3 中 t_1 时刻的概率攻击图 PAG_1 为例，分析在未观测到任何报警事件以及在观测到报警事件 $O = \{o_1, o_3\}$ 时，各节点的先验概率和后验概率的变化情况，如图 5 所示。

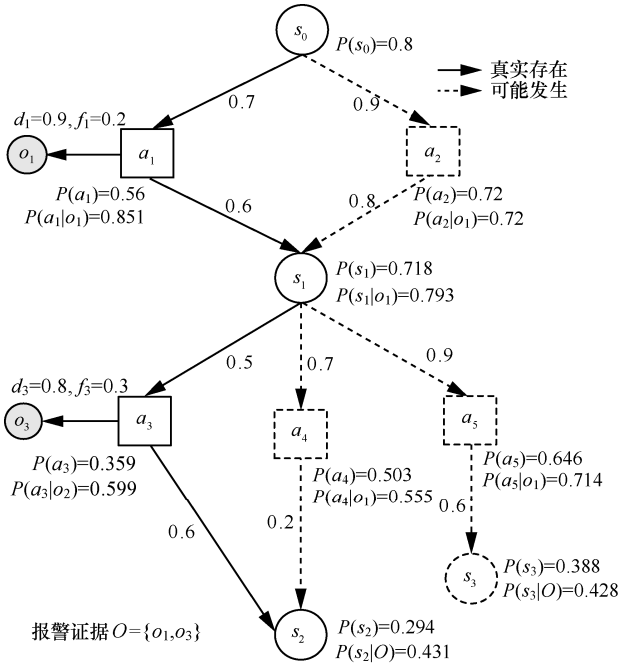

图 5 带观测节点的概率攻击图

通过定性与定量分析可知以下 3 点成立。

1) 在未观测到报警事件 o_1 和 o_3 时，攻击 a_2 发生的概率大于攻击 a_1 发生的概率，节点 s_3 被入侵的概率大于节点 s_2 被入侵的概率，攻击者的攻击意图为 s_3，攻击路径为 $\mathrm{path} = \{s_0, a_2, s_1, a_5, s_3\}$。

2) 在观测到报警事件 o_1 和 o_3 时，攻击 a_1 和 a_3 发生的概率提高了，其后验概率皆大于其先验概率，且后验概率皆大于 50%，表明 o_1 和 o_3 是真实报警。

3) 报警 o_1 和 o_3 为真实有效报警，因此攻击者的攻击意图为 s_2，s_2 被入侵的概率提高到 0.431，攻击路径为 $\mathrm{path} = \{s_0, a_1, s_1, a_3, s_2\}$。

算法 2 首先采用广度优先算法计算各个节点的先验概率和后验概率，将不具有子节点的状态节点加入目标节点集 G 中；如果某目标状态节点对应的动作节点存在真实报警，则该目标状态为攻击意图；否则对目标状态节点的后验概率进行比较排序，具有最大后验概率 $g_{\max}$ 的目标节点即为攻击意图，该算法同时可以识别有效报警证据。

算法 2 攻击意图推断算法

输入 候选攻击序列集合 $H = \{\mathrm{path}^k(s_0, G)\}$，观测事件序列 $O = \{o_i \mid_{i=1}^{m}\}$， $P(s_0)$

输出 节点先验概率 v_i.Pr 和后验概率 v_i.Po，攻击意图 G_{MAP}

1) begin
2) initialization queue Q
3) queue_push (Q, s_0) //将初始节点压入队列中
4) while (Q is not empty) do
5) { c_i = queue_pop (Q)
6) for each v_i is child of c_i do
7) { queue_push (Q, c_i)
8) if ($v_i \in A$) then //动作节点的 Pr 和 Po
9) { $v_i.\mathrm{Pr} = \prod \mathrm{Pre}(v_i).\mathrm{Pr}\Delta_s$
10) if (o_i exist) then //存在报警，计算置信度
11) $v_i.\mathrm{Po} = \dfrac{d_i v_i.\mathrm{Pr}}{d_i v_i.\mathrm{Pr} + f_i(1 - v_i.\mathrm{Pr})}$
12) if (v_i.Po>50% && v_i.Po>v_i.Pr) then
13) o_i is true //有效报警
14) else
15) o_i is false //误警，攻击 v_i 并未发生
16) $v_i.\mathrm{Po} = \prod \mathrm{Pre}(v_i).\mathrm{Po}\Delta_a$
17) end if
18) else if (o_i not exist) then
19) $v_i.\mathrm{Po} = \prod \mathrm{Pre}(v_i).\mathrm{Po}\Delta_a$ }
20) end if

21)　else if ($v_i \in S$) then //状态节点的 Pr 和 Po

22)　$v_i.\mathrm{Pr} = 1 - \prod(1 - \mathrm{Pre}(v_i).\mathrm{Pr}\Delta_a)$

23)　$v_i.\mathrm{Po} = 1 - \prod(1 - \mathrm{Pre}(v_i).\mathrm{Po}\Delta_a)$ }

24)　end if

25)　end for

26) } end while

27) return v_i.Pr, v_i.Po//得到所有节点的 Pr 和 Po

28) if ($v_i \in S$ && v_i has no child)

29)　add v_i to G //v_i 为目标状态节点

30) end if

31) if ($g \in G$　&& Pre(g) exist o_i)

32)　return g

33) else if ($g \in G$　&&　$\forall$Pre(g)　not exist o_i)

34)　return　$g_{\max} = \max(G.\mathrm{Po})$　//具有　$g_{\max}$ 的节点

35) end if

复杂度分析。算法 2 的复杂度与图广度搜索算法相同，为候选攻击序列集合中状态节点数、动作节点数以及有向边数之和，即 $O(\mathrm{num\_}S + \mathrm{num\_}A + \mathrm{num\_}E)$。

4.2.3　最大概率攻击路径推断

最大概率攻击路径推断是计算攻击者发动攻击行为的整个可能性，从候选攻击序列集合中选择具有最大后验概率 MAP 的攻击路径，即 $h_{\mathrm{MAP}} = \arg\max_{h_i \in H} P(h_i \mid O, \mathrm{PO})$。

在算法 2 得到所有节点的先验概率 $P(v_i)$和后验概率 $P(v_i|o_i)$的基础上，算法 3 以攻击意图为起点，通过贪心反向查找具有最大后验概率的动作节点 a_i 和状态节点 s_i，即可得到最大概率攻击路径 h_{MAP}。如果当前节点为状态节点，则查找具有真实报警的前置动作节点并加入攻击路径 h_{MAP} 中；而当出现漏报不存在报警时，则查找后验概率最大的动作节点并加入路径 h_{MAP} 中；如果当前节点为动作节点，则将其所有的前置状态节点都加入路径 h_{MAP} 中。

算法 3　最大概率攻击路径推断算法

输入　攻击意图 G_{MAP}，节点先验概率 v_i.Pr 和后验概率 v_i.Po

输出　$h_{\mathrm{MAP}} = (s_0 \to \cdots s_i \to \cdots G)$。

1) begin

2) h_{MAP} = {}

3) initialization queue Q

4) add G_{MAP} to h_{MAP} //将 G_{MAP} 加入路径 h_{MAP}

5) queue_push (Q, G_{MAP}) //将 G_{MAP} 加入队列

6) while (Q is not empty) do

7) {　v_i = queue_pop (Q)

8)　if ($v_i \in S$) then //当前节点为状态节点

9)　{ if ($\exists a_i \in \mathrm{Pre}(v_i)$ && o_i exist) then //其前置节点a_i存在报警，说明a_i已发生并将a_i加入路径h_{MAP}

10)　add a_i to h_{MAP}

11)　queue_push (a_i)

12)　else if ($\exists a_i \in \mathrm{Pre}(v_i)$ && o_i not exist) then

13)　$\max_{a_i \in \mathrm{Pre}(v_i)}(a_i.\mathrm{Po})$　//其前置节点 a_i 产生漏报，将具有 MAP 的 a_i 加入 h_{MAP}

14)　add a_i to h_{MAP}

15)　queue_push (a) }

16)　else if ($v_i \in A$) then　//当前节点为动作节点

17)　add Pre(v_i) to h_{MAP}　//将其所有的前置状态节点都加入 h_{MAP}

18)　queue_push (Pre(v_i))

19) } end while

20) return h_{MAP}

复杂度分析。算法 3 的复杂度与算法 2 相似，但由于已经确定了攻击意图，状态节点数、动作节点数相对算法 2 减少了，且与有向边无关，故复杂度为 $O(\mathrm{num\_}S + \mathrm{num\_}A)$。

4.3　攻击场景演化

通过上述报警映射、攻击意图推断和最大概率攻击路径计算，能够得到准确可信的攻击场景。然而由于云环境弹性、动态性的特点，其概率攻击图需要周期性地动态更新，攻击场景也应周期性地动态演化，从而保证攻击场景的完备性和新鲜性。

定义 12　动态攻击场景（DAS, dynamic attack scenario）。DAS 指在时间域 T=[t_0, t_n]内随攻击步时 Δt 动态演化的攻击场景序列，可表示为

$$\mathrm{DAS}^T = \{< \mathrm{AS}_i, t_i > \mid 0 \leqslant i \leqslant n, t_i = t_{i-1} + \Delta t\}$$

$$\mathrm{AS}_i : \mathrm{map}(O_i, \mathrm{PAG}_i)$$

其中，二元组 $< \mathrm{AS}_i, t_i >$ 表示在 t_i 时刻上的攻击场景 AS_i，O_i 和 PAG_i 分别表示 t_i 时刻观测到的新报警序列和更新得到的概率攻击图，将两者进行映射 map 即可得到 t_i 时刻的攻击场景 AS_i。

攻击场景演化过程如图 6 所示。由图 6 可知，将第一个攻击周期 Δt 内的报警序列 O_1 与 t_1 时刻上

的概率攻击图 PAG_1 相映射，映射成功则得到 t_1 时刻上的攻击场景 AS_1；同样将第二个 Δt 内的报警序列 O_2 与 t_2 时刻上的攻击图 PAG_2 相映射，得到攻击场景 AS_2；以此类推，可得到 t_3 时刻上的攻击场景 AS_3。由此可知，时间域 T 内动态演化的攻击场景序列为 $DAS^T=\{<AS_1,t_1>,<AS_2,t_2>,<AS_3,t_3>\}$。根据攻击场景 DAS^T 可以发现，攻击者在 t_1 时刻向目标主机 $host_1$ 发动了攻击行为 a_1，获得了 $host_1$ 的 user 权限；在 t_2 时刻进一步向 $host_1$ 发动了攻击 a_3，获得 $host_1$ 的 root 访问权限；在 t_3 时刻通过 $host_1$ 获得了 $host_1$ 的访问权限，t_3 时刻的攻击场景 AS_3 描述了攻击过程的整体叠加情况。可见，动态演化的攻击场景能够完整地还原攻击者的逐步渗透过程与权限提升过程，从整体上把握云环境的安全态势。

5　实验验证

本节首先介绍云平台实验环境以及此环境中的安全漏洞，构建该环境下的初始概率攻击图；然后，根据观测报警序列及其置信度推断出攻击者的攻击意图和最大概率攻击路径，并构建动态演化的攻击场景。

5.1　实验环境

云平台实验环境如图 7 所示。云平台中包括了 4 个浪潮云服务器，搭建了 Xen 虚拟化云系统，配置信息如表 4 所示。

表 4　　云平台配置信息

软硬件	版本
CPU	2 颗 Intel E5-2620v3 (2.4 GHz/6c) 8GT/15ML3
内存、硬盘	96 GB、900 GB
虚拟化系统	Xen 4.6.0, Ubuntu 16.04

云服务器 PM_1 被分割成了 4 个 VM，分别运行网络控制器（NC, network control）、Nessus 漏洞扫描器、MulVAL 攻击图生成工具以及 IDS；云服务器 PM_2 被

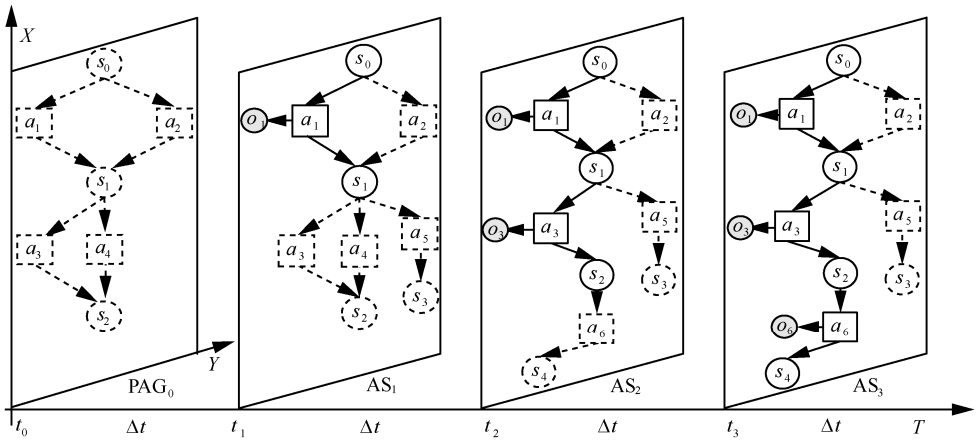

图 6　攻击场景演化过程

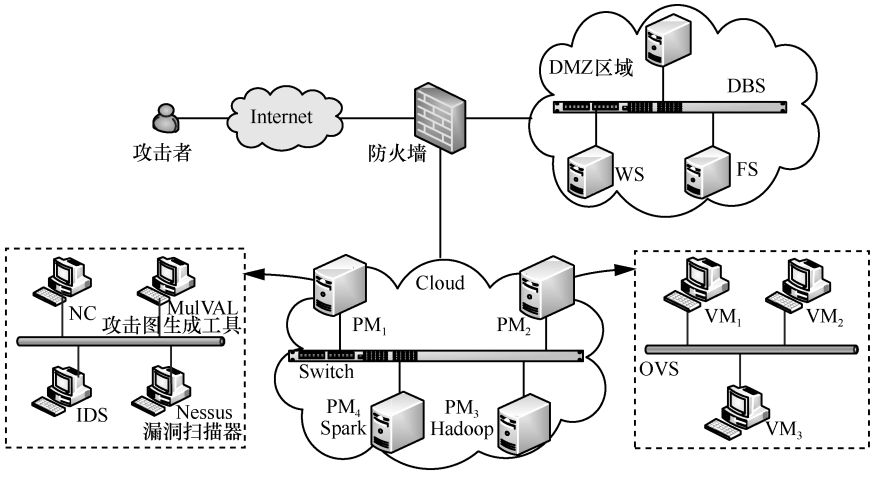

图 7　云平台实验环境

分割成若干个 VM，作为云租户所用 VM，VM 间由可编程的虚拟交换机（OVS, open vSwitch）相连接。OVS 仅完成数据转发功能，而路由控制则由 PM_1 中 NC 来完成；服务器 PM_3 和 PM_4 分别运行 Hadoop 和 Spark 分布式计算平台。在 DMZ 区域部署了网站服务器（WS, Web server）、文件服务器（FS, file server）等。这里假设 PM_3 只允许 VM_1 访问其计算服务，而 PM_4 只允许 VM_2 访问其服务。

5.2　实验结果分析与比较

5.2.1　概率攻击图的结构和参数

利用 Nessus 扫描器对云平台中各主机进行漏洞扫描，获得的各漏洞信息如表 5 所示。其中，Xen 系统的漏洞 CVE-2017-2620 为 t_i 时刻新发现的漏洞。将网络连通性、网络配置及漏洞信息等输入 MulVAL 工具，从而生成初始概率攻击图 PAG_0 的网络结构，如图 8 所示。

从图 8 可知，存在 6 种可能的攻击路径，其中，攻击者利用 WS 上的溢出漏洞 CVE-2010-2227 获取 WS 的管理员权限，进而通过 NFS 接口在 FS 上写入木马程序，该木马程序被不小心执行，从而让攻击者控制了 FS。当然攻击者也可以直接利用 FS 上的溢出漏洞 CVE-2017-7895 获取 FS 的权限。接着，VM_2 从 FS 上下载了带木马的文件从而被控制，攻击者操控 VM_2 利用漏洞 CVE-2018-8024 对 Spark 服务器发动攻击。同样，攻击者操控 FS 对虚拟机 VM_1 发动攻击，并进一步利用漏洞 CVE-2016-3086 对 Hadoop 平台展开攻击，窃取其重要信息或文件等。

根据图 8 中概率攻击图的结构和表 5 中 CASS 属性（AV/AC/Au）能够确定概率攻击图 PAG_0 中有向边的权值，如表 6 所示。表 6 中有向边(s_8, a_{12})和(a_{12}, s_9)为 t_i 时刻发现新漏洞时新增的有向边。依据网络结构和网络参数，构建完整的初始概率攻击图 PAG_0。

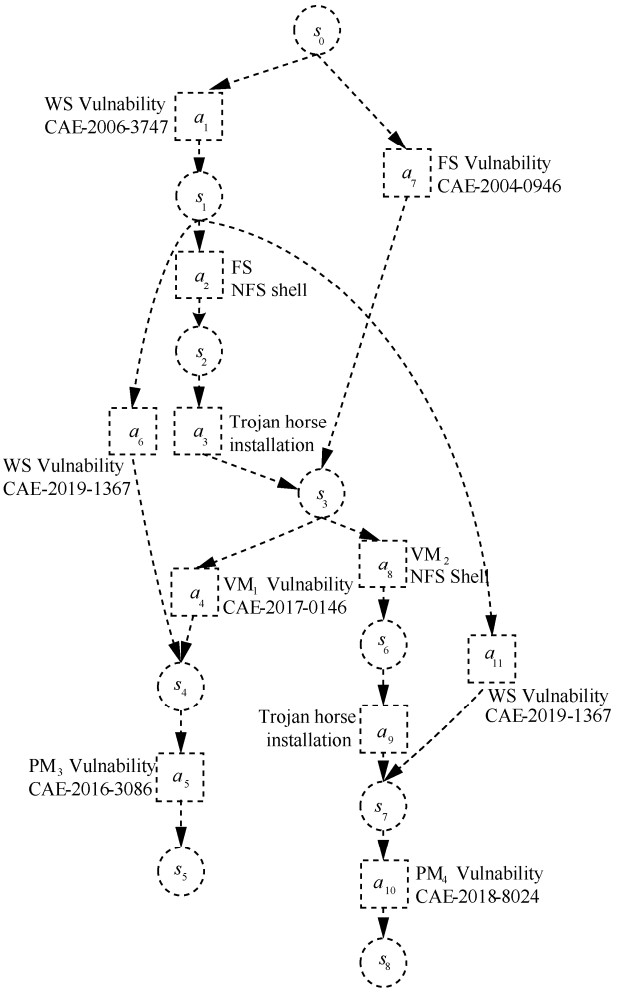

图 8　初始概率攻击图的结构

表 5　　**云平台漏洞信息**

主机	版本	漏洞	描述	AV/AC/Au
WS	Apache 2.0	CVE-2006-3747	execute code, dos	N/H/N
WS	IE 9	CVE-2019-1367	overflow, execute code	N/H/N
FS	NFS 1.0	CVE-2004-0946	execute code, overflow	N/L/N
VM_1	Win7 SP1	CVE-2017-0146	execute code	N/M/N
PM_3	Hadoop 2.6	CVE-2016-3086	obtain information	N/L/N
PM_4	Spark 2.1	CVE-2018-8024	obtain infotmation	N/M/S
VMM	Xen 4.6	CVE-2017-2620	dos, gain privileges	L/M/N

表 6 有向边的权值

有向边	权值	有向边	权值	有向边	权值	有向边	权值
(s_0, a_1)	0.49	(s_3, a_4)	0.86	(s_0, a_7)	1.0	(s_7, a_{10})	0.68
(a_1, s_1)	1.0	(a_4, s_4)	1.0	(a_7, s_3)	1.0	(a_{10}, s_8)	1.0
(s_1, a_2)	0.5	(s_4, a_5)	1.0	(s_3, a_8)	0.5	(s_1, a_{11})	0.49
(a_2, s_2)	0.9	(a_5, s_5)	1.0	(a_8, s_6)	0.9	(a_{11}, s_7)	1.0
(s_2, a_3)	0.5	(s_1, a_6)	0.49	(s_6, a_9)	0.5	(s_8, a_{12})	0.34
(a_3, s_3)	0.9	(a_6, s_4)	1.0	(a_9, s_7)	0.9	(a_{12}, s_9)	1.0

5.2.2 攻击场景构建过程及分析

在初始概率攻击图 PAG_0 的基础上，依据攻击步时 Δt 内的更新操作情况，利用算法 1 生成概率攻击图序列。依据 IDS 观测到的报警序列 O，利用算法 2 和算法 3 即可推断出攻击意图 g_{MAP} 和最大概率攻击路径 h_{MAP}，从而构建出攻击场景，如表 7 所示。

概率攻击图的动态更新及攻击场景的动态演化过程如图 9 所示。为了不失一般性，在 t_0 时刻假设没有观测到任何报警事件 $O_0 = \{\mathrm{null}\}$，这相当于初始概率攻击图 PAG_0。图 9(a)直观地展示了攻击者最可能的攻击意图和攻击路径。在该情况下，状态节点 s_5 的概率是 0.89，状态节点 s_8 的概率是 0.27，因此攻击者的攻击意图为 s_5，攻击目标是运行 Hadoop 的主机 PM_3，最大概率攻击路径则是$\{s_0, a_7, s_3, a_4, s_4, a_5, s_5\}$。

在 t_1 时刻更新概率攻击图，假设 $\mathrm{GC}_1 = \varnothing$，即攻击图没有任何更新操作，IDS 在 t_1 时刻观测到报警事件 $O_1 = \{o_1, o_2, o_3\}$，其对应的置信度为 $\mathrm{PO} = \{0.81, 0.6, 0.38\}$。在此情况下可知，报警 o_3 为误报警，攻击 a_3 实际上并未发生，其发生的概率应是 0.27。候选攻击序列包括$\{s_0, a_1, s_1, a_2, s_2, a_3, s_3, a_8, s_6, a_9, s_7, a_{10}, s_8\}$和$\{s_0, a_1, s_1, a_2, s_2, a_3, s_3, a_4, s_4, a_5, s_5\}$，经计算知状态节点 s_5 的概率是 0.92，状态节点 s_8 的概率是 0.35，因此攻击者的攻击意图仍为 s_5，最大概率攻击路径为$\{s_0, a_1, s_1, a_2, s_2, a_3, s_3, a_4, s_4, a_5, s_5\}$，攻击场景如图 9(b)所示。

在 t_2 时刻修复了 PM_3 中的漏洞，则在概率攻击图中应删除对应的失效动作节点 a_5 及其后置状态节点 s_5 以及相关边，即执行更新图操作 $\mathrm{GC}_2 = \{< D, a_5, s_5 >\}$。假设 t_2 时刻观测到报警事件 $O_2 = \{o_{11}, o_{10}\}$，其对应的置信度为 $\mathrm{PO} = \{0.59, 0.62\}$，报警事件 O_2 皆为有效报警，则可判断目标节点为 s_5，攻击意图为 s_8，最大概率攻击路径为$\{s_0, a_1, s_1, a_{11}, s_7, a_{10}, s_8\}$，如图 9(c)所示。需要说明的是，由于在 t_1 时刻攻击者成功发动了攻击 a_1 和 a_2，已经获得了资源或权限能力，虽然在 t_2 时刻没有观测到报警 o_1 和 o_2，但攻击 a_1 和 a_2 发生的概率将保持不变。

在 t_3 时刻发现 Xen 系统中存在新的漏洞 CVE-2017-2620，则在概率攻击图中需要增加与漏洞相对应的动作节点 a_{12}，并基于可达性关系搜索该原子攻击的前置状态。由于虚拟机 VM_2 能够访问 Xen 系统的虚拟 CPU、虚拟内存等资源，可能发动 DoS 或漏洞利用攻击，因此节点 a_{12} 的前提状态为 VM_2 的状态 s_7，通过前提边进行连接，同时增加该动作节点的后置状态 s_9 和后果边，更新操作为 $\mathrm{GC}_3 = \{I, a_{12}, s_9\}$。同样，在 t_3 时刻观测到报警事件 o_{12}，其对应的置信度为 0.54，则可判断出 t_3 时刻的攻击意图为 s_9，最大概率攻击路径为$\{s_0, a_1, s_1, a_{11}, s_7, a_{10}, s_8, a_{11}, s_9\}$，如图 9(d)所示。此时的攻击场景包括两条成功的攻击路径和一次失败的攻击尝试，展现了攻击者的攻击渗透过程。

表 7 攻击场景构建过程

时刻	更新操作	编号	观察事件序列	攻击意图 g_{MAP}	最大概率攻击路径 h_{MAP}
t_0	—	O_0	{null}	s_5	$\{s_0, a_7, s_3, a_4, s_4, a_5, s_5\}$
t_1	$\varnothing$	O_1	{o_1:0.81, o_2:0.6, o_3:0.38}	s_5	$\{s_0, a_1, s_1, a_2, s_2, a_3, s_3, a_4, s_4, a_5, s_5\}$
t_2	$\{D, a_5, s_5\}$	O_2	{o_{11}:0.59, o_{10}:0.62}	s_8	$\{s_0, a_1, s_1, a_{11}, s_7, a_{10}, s_8\}$
t_3	$\{I, a_{12}, s_9\}$	O_3	{o_{12}:0.54}	s_9	$\{s_0, a_1, s_1, a_{11}, s_7, a_{12}, s_9\}$

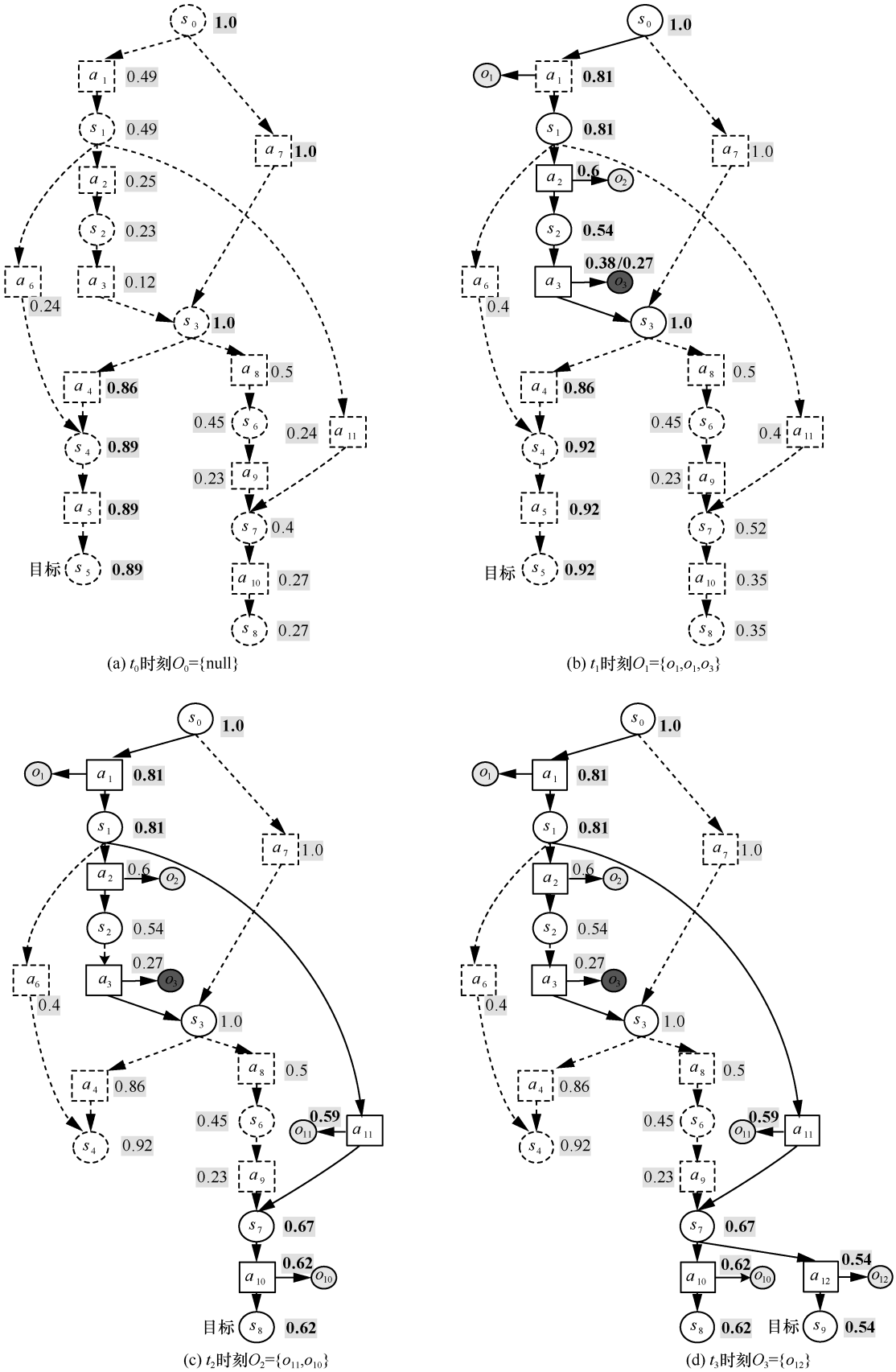

图 9　概率攻击图动态更新及攻击场景动态演化过程

5.2.3 实验比较分析

文献[10]和文献[13]的工作中没有考虑报警观测事件推导到攻击发生的不确定性，观测事件与攻击发生是一一对应的，即报警置信度 PO 的取值全为 1，从而无法识别真实有效的报警。文献[11]的工作中没有解决误报、漏报问题对攻击场景造成的影响。上述方法均没有考虑攻击场景的动态演化，无法保证攻击场景的完备性和新鲜性。表 8 展示了现有基于概率攻击图的报警关联方法在各方面的对比。从表 8 可以看出，本文方法支持有效报警识别和动态演化，优于其他 3 种方法。

表 8 基于概率攻击图的报警关联方法对比

方法	攻击意图推断	攻击路径推断	误报漏报处理	有效报警识别	动态演化
文献[10]方法	√	√	√	—	—
文献[12]方法	√	√	—	√	—
文献[13]方法	√	√	√	—	—
本文方法	√	√	√	√	√

实验验证了本文方法的有效性。此外，本文方法具有以下优势。

1) 通过将报警事件与概率攻击图进行映射匹配，从定性与定量分析 2 个方面构建了攻击场景，不仅能够解决误报、漏报问题，识别有效报警证据，还能够推断出攻击者的攻击意图和实际攻击路径，从而保证了攻击场景的准确性。

2) 动态概率攻击图是随时间推移而动态变化的，依据动态概率攻击图，将攻击场景随之动态演化，能够适应云计算环境弹性、动态性的特点，保证了攻击场景的完备性和新鲜性。

3) 全时域内的动态攻击场景完整地展现了攻击者的逐步渗透和权限提升过程，方便管理员从整体上把握云环境的安全态势，确定攻击演变过程中危害较大的阶段或者关键节点，进而采取切实有效的措施来抑制攻击的发生。

6 结束语

面向云环境的攻击场景重构技术逐渐成为网络安全防御领域的研究热点，旨在再现攻击者的逐步渗透过程和攻击活动全貌，是应对复杂多步攻击的重要手段。本文在概率攻击图的基础上，立足于攻击步骤间的因果依赖关系和概率推理关系，考虑其时空动态特性，创建了动态概率攻击图模型，设计了概率攻击图更新算法，实现了概率攻击图的周期性或触发性更新，解决了传统的静态不变的概率攻击图无法适应弹性、动态性的云环境问题；利用概率攻击图作为模板，将异常检测系统观测到的报警与概率攻击图进行映射匹配，能够获得初步的攻击场景；针对误报、漏报等现象导致攻击场景存在错误、歧义、断裂等不确定性问题，设计了攻击意图推断算法和最大概率攻击路径推断算法，利用最大后验估计原理对候选攻击序列进行定量推理，重构出准确可信的攻击场景；同时将攻击场景随动态概率攻击图动态演化，保证了攻击场景的完备性和新鲜性。通过攻击场景的初建、重构及演化，实现了云环境复杂多步攻击的及时发现。实验结果表明，本文方法能够有效地推断出攻击意图和计算攻击路径，构建完整的、动态演化的攻击场景，进而帮助云安全管理员从整体上把握云环境的安全态势，为构建可监管可追责的云环境提供了一定的依据和参考。

参考文献：

[1] PETER M M, TIMOTHY G. SP 800-145. The NIST definition of cloud computing[M]. National Institute of Standards & Technology, 2011.

[2] PENG N, YUN C, REEVES D S, et al. Constructing attack scenarios through correlation of intrusion alerts[C]//ACM Symposium on Computer and Communications Security. New York: ACM Press, 2002: 245-254.

[3] WANG L, GHORBANI A, LI Y, et al. Automatic multi-step attack pattern discovering[J]. International Journal of Network Security, 2010, 10(2): 142-152.

[4] 梅海彬, 龚俭, 张明华, 等. 基于警报序列聚类的多步攻击模式发现研究[J].通信学报, 2011, 32(5): 63-69.
MEI H B, GONG J, ZHANG M H, et al. Research on discovering multi-step attack patterns based on clustering IDS alert sequences[J]. Journal on Communications, 2011, 32(5): 63-69.

[5] 葛琳, 季新生, 江涛, 等. 基于关联规则的网络信息内容安全事件发现及其 Map-Reduce 实现[J]. 电子与信息学报, 2014, 36(8): 1831-1837.
GE L, JI X S, JIANG T, et al. Association rules and its implementation in Map-Reduce[J]. Journal of Electronics & Information Technology, 2014, 36(8): 1831-1837.

[6] 鲁显光, 杜学绘, 王文娟, 等. 基于改进 FP growth 的告警关联算法[J]. 计算机科学, 2019, 46(8): 64-70.
LU X G, DU X H, WANG W J, et al. Alert correlation algorithm based on improved FP growth[J]. Computer Science, 2019, 46(8): 64-70.

[7] WANG S, TANG G, KOU G, et al. An attack graph generation method based on heuristic searching strategy[C]//IEEE International Conference on Computer and Communications. Piscataway: IEEE Press, 2016: 1180-1185.

[8] KAYNAR K, SIVRIKAYA F. Distributed attack graph generation[J]. IEEE Transactions on Dependable and Secure Computing, 2016, 13(5): 519-532.

[9] 吕慧颖, 彭武, 王瑞梅, 等. 基于时空关联分析的网络实时威胁识别与评估[J]. 计算机研究与发展, 2014, 51(5): 1039-1049.
LYU H Y, PENG W, WANG R M, et al. A real-time network threat recognition and assessment method based on association analysis of time and space[J]. Journal of Computer Research and Development, 2014, 51(5): 1039-1049.

[10] 刘威歆, 郑康锋, 武斌, 等. 基于攻击图的多源告警关联分析方法[J]. 通信学报, 2015, 36(9): 135-144.
LIU W X, ZENG K F, WU B, et al. Alert processing based on attack graph and multi-source analyzing[J]. Journal on Communications, 2015, 36(9): 135-144.

[11] 陈小军, 方滨兴, 谭庆丰, 等. 基于概率攻击图的内部攻击意图推断算法研究[J]. 计算机学报, 2014, 37(1): 62-72.
CHEN X J, FANG B X, TAN Q F, et al. Inferring attack intent of malicious insider based on probabilistic attack graph[J]. Journal of Computers, 2014, 37(1): 62-72.

[12] 王硕, 汤光明, 王建华, 等. 基于因果知识网络的攻击场景构建方法[J]. 计算机研究与发展, 2018, 55(12): 2620-2636.
WANG S, TANG G M, WANG J H, et al. Attack scenario construction method based on causal knowledge net[J]. Journal of Computer Research and Development, 2018, 55(12): 2620-2636.

[13] 许嘉, 张千桢, 赵翔, 等. 动态图模式匹配技术综述[J]. 软件学报, 2018, 29(3): 663-688.
XU J, ZHANG Q Z, ZHAO X, et al. Survey on dynamic graph pattern matching technologies[J]. Journal of Software, 2018, 29(3): 663-688.

[14] OU X, GOVINDAVAJHALA S, APPEL A W, et al. MulVAL: a logic-based network security analyzer[C]//14th USENIX Security. Berkeley: USENIX Association, 2005: 1-16.

[15] JAJODIA S, NOEL S. Topological vulnerability analysis: a powerful new approach for network attack prevention, detection, and response[J]. Algorithms, Architectures and Information Systems Security, 2005: 285-305.

[16] LIPPMANN R, INGOLS K, SCOTT C, et al. Validating and restoring defense in depth using attack graphs[C]//Milcom 2006 Military Communications Conference. [S.n.:s.l.], 2006: 1-10.

[17] SCARFONE K, MELL P. An analysis of CVSS version 2 vulnerability scoring[C]//International Symposium on Empirical Software Engineering & Measurement. Piscataway: IEEE Press, 2009.

[18] 冯学伟, 王东霞, 黄敏桓, 等. 一种基于马尔可夫性质的因果知识挖掘方法[J]. 计算机研究与发展, 2014, 51(11): 2493-2504.
FENG X W, WANG D X, HUANG M H, et al. A mining approach for causal knowledge in alert correlating based on the Markov property[J]. Journal of Computer Research and Development, 2014, 51(11): 2493-2504.

[作者简介]

王文娟（1981− ），女，河南鹤壁人，信息工程大学博士生、副教授，主要研究方向为网络与信息安全、云计算安全。

杜学绘（1963− ），女，河南新乡人，博士，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、云计算安全、大数据安全等。

单棣斌（1983− ），男，河北邯郸人，信息工程大学博士生、讲师，主要研究方向为网络与信息安全、大数据安全。