·安全技术· 文章编号：1000-3428(2018)08-0184-08 文献标志码：A 中图分类号：TP393.08

基于时间增益补偿率的攻击路径优化算法

王 辉，王银城，鹿士凯

（河南理工大学 计算机科学与技术学院，河南 焦作 454000）

摘 要： 现有网络攻击模型由于多数未考虑时间成本，导致攻击路径节点置信度计算不准确。为此，通过对攻击路径进行攻击成本权重分析，提出一种基于时间成本的攻击路径生成算法，从而消除冗余路径。通过对冗余路径节点进行标识，避免节点置信度计算不准确的问题。实验结果表明，该算法能够有效地消除冗余路径，提高节点置信度计算的准确性，实现攻击路径的优化。

关键词： 节点置信度；时间成本；时间增益补偿率；似然加权；路径优化

中文引用格式： 王 辉，王银城，鹿士凯. 基于时间增益补偿率的攻击路径优化算法[J]. 计算机工程，2018，44(8)：184-191，198.

英文引用格式： WANG Hui，WANG Yincheng，LU Shikai. Attack path optimization algorithm based on time gain compensation rate[J]. Computer Engineering，2018，44(8)：184-191，198.

Attack Path Optimization Algorithm Based on Time Gain Compensation Rate

WANG Hui，WANG Yincheng，LU Shikai

（School of Computer Science and Technology，Henan Polytechnic University，Jiaozuo，Henan 454000，China）

【Abstract】 The existing network attack model is not accurate because it does not consider the time cost. Therefore，an attack path generation algorithm based on time cost is proposed to eliminate redundant paths by analyzing the cost weight of attack path. Through the identification of redundant path nodes，the inaccuracy of node confidence calculation is avoided. Experimental results show that the algorithm can effectively eliminate redundant paths，improve the accuracy of node confidence calculation and achieve the optimization of attack path.

【Key words】 node confidence；time cost；time gain compensation rate；likelihood weighting；path optimization

DOI： 10.19678/j.issn.1000-3428.0047937

0 概述

随着互联网规模的不断扩大，拓扑结构日益复杂，攻击行为逐渐呈现出智能化的特点，攻击工具由被动发动攻击发展到主动发起攻击，并且能够根据环境变化或设计规则自我调整。攻击行为的破坏性报告显示：2016 年 CNCERT/CC 发现恶意程序数量相比于 2015 年增长 39.0%；网站后门增长 9.3%；2016 年 CNVD 收录漏洞数量较 2015 年增长 33.9%；工业控制漏洞增长 38.4%[1]。上述数据表明，网络攻击对我国经济社会发展和国家安全造成了严重威胁，因此，加强网络攻击研究，寻找切实有效的安全防御措施乃是当前工作的重中之重。

网络攻击行为很大程度上是一种主观行为，具有一定的不确定性[2-3]，网络攻击图具有处理不确定关系的能力[4-5]，因此，如何采用攻击图进行攻击行为预测，成为国内外研究的焦点。

文献[6]提出用攻击图建模的方法来预测网络中存在的潜在攻击路径的方法；文献[7]提出网络攻击图中存在冗余的攻击行为，认为能够通过某种方式消除该类攻击行为；文献[8]提出网络攻击图中存在概率计算的不确定性，认为该类不确定性是可以计算的；文献[9]提出网络攻击中存在节点转移，且该类转移是可以预测的；文献[10]提出一种基于反取证的攻击图分析方法，通过对路径偏差的各种可能性分析，建议相应的防御策略。

文献[11]提出一种基于脆弱性攻击图的并行分布式计算方法，认为可以通过散布在目标网络的传感器收集到的入侵警报数据来实现攻击图的计算。但是当攻击者试探性攻击时仍可能被检测到，由此得来的冗余数据仍会被用于计算，使得数据量较为庞大，同时影响节点置信度的计算。

文献[12]提出了一种量化多步攻击的安全风险评估方法，针对攻击图中概率计算错误的问题，提出

基金项目： 国家自然科学基金(61300216)。

作者简介： 王 辉(1975—)，男，副教授、博士，主研方向为网络安全；王银城、鹿士凯，硕士研究生。

收稿日期： 2017-07-13 **修回日期：** 2017-08-30 **E-mail：** wanghui_jsj@hpu.edu.cn

最大可达概率的计算方法,该方法在攻击路径的优化方面具有一定的可行性,然而却忽略了作为智能主体的攻击者在选择攻击时,不仅要考虑可达概率,还要考虑攻击成本。

文献[13]提出了一种基于攻击距离(即攻击复杂度序列求和)的攻击图优化算法,通过该方法在一定程度上能够消除冗余路径。但是文章只考虑了攻击复杂度,而没有考虑作为智能主体的攻击者在发动攻击时不仅要考虑攻击复杂度,还要考虑时间复杂度。

文献[14]在攻击图中引入条件概率表,利用安全监控系统给出的观测事件的置信度来计算攻击意图的概率。但是当攻击者试探性攻击时,仍会被安全监控系统检测到,导致节点置信度的错误计算。

文献[15]提出了一种基于攻击代价的选择性攻击方法,该方法试图通过攻击者的角度消除路径冗余,由于文章只考虑了操作成本,却忽视了作为智能主体的攻击者不仅要考虑操作成本,还要考虑时间成本。

为解决上述问题,本文提出一种基于网络攻击图的攻击路径优化模型,在该模型中引入时间成本的概念,并综合各方面的影响因素得出攻击成本的计算公式,通过对其余因素的权重进行分析得出时间成本与攻击成本的相关程度,以及时间成本与可达概率的相关程度,给出基于时间成本的时间增益补偿率的攻击可行性分析。将攻击者对攻击路径的选择转换为对时间成本可行性值的判断,并以此为基础得出基于时间增益补偿率的攻击路径优化算法。

1　贝叶斯攻击图定义以及攻击路径描述

网络攻击是单步攻击有机组合的过程,它包含一系列与资源相关的基本攻击行为,攻击者运用基本攻击行为不断以当前获取的资源为跳板获取下一个资源,最终占有目标资源,从主观上来说,网络攻击是一种时间行为,因为随着攻击时间增加,攻击者面临的压力成倍增加。

网络攻击图主要综合上述 3 种因素(资源状态、攻击行为、时间成本)及相互关系提出的一种描述网络安全状态的表示方法。它包含网络攻击所有可能的攻击路径以及引起资源状态变迁的攻击序列,根据单调性假设,攻击者不会放弃已有的资源和收益,也不会试图去获取已经占有的资源和收益。此外,当攻击者认为对某一目标节点的攻击行为所付出的时间成本大于所能承受的范围时,就放弃攻击,因此,网络管理员可以对攻击行为进行时间成本分析,推断攻击者可能选择的攻击路径,并及时采取相应的防护措施,根据以上分析定义如下网络攻击图、攻击路径。

定义 1(网络攻击图) $BNWAG=(V,E,P,S,\Pi,D,T,W)$为一个有向无圈图。

1) $V=V_0\cup V_G$为资源状态节点集合。V_0 为攻击者已成功获取的初始资源状态集合,$V_G\subseteq V$ 表示攻击者试图获取的目标资源状态集合,单个资源状态节点用 v_j表示,v_j的前驱节点用 $Pre(v_j)$表示,后继节点用 $Post(v_j)$表示。

2) $E=\{e_{i\to j}\mid i=1,2,\cdots,N,j=1,2,\cdots,N\}\mid i=1,2,\cdots,N,j=1,2,\cdots,N\}$ 为资源状态节点之间存在的有向边集合即攻击行为集合。$e_i\to_j$表示单个有向边,$\forall e_{i\to j}$都存一条 $<v_i,v_j>$的有向边。

3) P 为概率集合,$P(v_j)$表示资源状态节点 v_j 的可达概率,$P(e_{i\to j})$表示攻击行为成功的概率。

4) $S=\{s(v_i)\mid i=1,2,\cdots,N)\}$ 为节点度集合,$s(v_i)$表示资源状态节点 v_i的节点度,详见定义 2。

5) $\Theta=\Theta_1\cup\Theta_2$ 为概率知识分布,Θ 依附于有向边 E 上。Θ_1 表示占有当前资源状态节点的前提下发生攻击行为的概率分布,即 $\Theta_1=(e_i\to_j=\text{True}\mid$占有资源状态节点 $v_i)\to(0,1]$。Θ_2 表示攻击行为发生的前提下成功占有下一个资源状态节点的概率分布即$\Theta_2=(e_{i\to j}$成功$\mid e_{i\to j}=\text{True})\to(0,1]$。

6) $D=\{\text{AND},\text{OR}\}$为资源状态节点之间逻辑关系集合。AND 表示资源状态节点之间存在“与”关系,OR 表示资源状态节点之间存在“或”关系。基本依赖关系如图 1 所示。

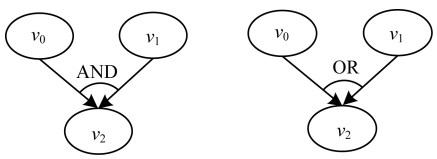


图 1　基本依赖关系

7) $T=\{t_{i\Lambda i+1\Lambda\cdots\Lambda i+n}\to_j\mid i=1,2,\cdots,N\}$ 为入侵状态转换关系的集合,$t_{i\Lambda i+1\Lambda\cdots\Lambda i+n}\to_j$表示从资源状态节点 $v_i,v_{i+1},\cdots,v_{i+n}$到资源状态节点 vj 的变迁,其中,$v_i,v_{i+1},\cdots,v_{i+n}$是存在 AND 关系的 v_j的前驱节点,若前驱节点之间不存在 AND 关系,用 $t_{i\to j}$表示,Λ 表示资源状态节点之间的 AND 关系。

8) W 为攻击权集合。$\forall w\in W$ 依附于有向边 E 上,由二元组(h,m)表示,h 表示攻击资源状态节点 v_i所花费的时间成本,m 表示时间成本临界值(依据专家经验得出)。

9) $I=\{V\cup E\}\to[0,1]$为网络攻击图节点置信度分布,$I(v_i)$表示已成功占有资源状态节点 v_i的概率,在起始条件下,由于资源状态节点 V_0已被攻击者占有,因此 $I(V_0)=1$。

根据以上定义,首先利用网络弱点扫描工具(如 nessus、nmap、X-scan)扫描系统资源存在的弱点、服

务等信息,把扫描得到的信息作为资源状态节点,用有向边标出节点之间的依赖关系,结合 AND-OR 节点的组合情况 OR-OR、AND-AND、OR-AND、AND-OR,生成目标系统的主干框架。并结合目标网络的安全状态,预测攻击者的攻击路径。依照上述方法,本文通过对小型局域网进行扫描,得到如图 2 所示的网络攻击图。

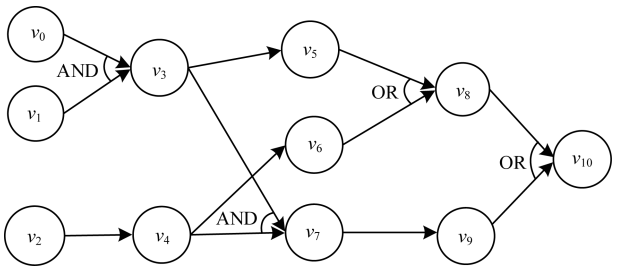


图 2 网络攻击图的一般形式

图 2 初始状态下攻击者以概率 $I(v_0)$、$I(v_1)$、$I(v_2)$占有初始资源状态节点 v_0、v_1、v_2后,发动攻击行为最终占有资源状态节点 v_{10}的过程,攻击行为 $e_{0\to3}$、$e_{1\to3}$、$e_{2\to4}$、$e_{3\to5}$、$e_{3\to7}$、$e_{4\to7}$、$e_{4\to6}$、$e_{5\to8}$、$e_{6\to8}$、$e_{7\to9}$、$e_{8\to10}$、$e_{9\to10}$的发生遵循 Θ_1 的条件概率分布,其后以条件概率分布 Θ_2 分别占有 v_3、v_4、v_5、v_6、v_7、v_8、v_9、v_{10}。资源状态节点 v_3、v_7发生的前提条件时要求攻击者必须分别同时占有资源状态节点 v_0、v_1以及 v_3、v_4,该攻击图发生了 10 次资源状态节点的变迁,即 $t_{0\wedge1\to3}$、$t_{2\to4}$、$t_{3\to5}$、$t_{4\to6}$、$t_{3\wedge4\to7}$、$t_{5\to8}$、$t_{6\to8}$、$t_{7\to9}$、$t_{8\to10}$、$t_{9\to10}$。

定义 2(节点度) $\forall v_j \in V, P(v_j) = P:(\mathrm{Pre}(v_j), v_j)\to[0,1]$,$\mathrm{Pre}(v_j)$满足条件的情况下发动攻击行为,成功占有资源状态节点 v_j 的难易程度。

本文引入节点度的概念,用来表示节点被攻破的难易程度,一方面它可以忽略内部复杂的结构,另一方可以表现出内部操作的复杂程度。

定义 3(攻击路径) 对于初始资源状态节点 $\forall v_i \in V_G$,依赖于初始资源状态节点 v_i 的攻击行为满足条件,成功占有资源状态节点 v_{i+1},如此反复,直至占有最终目标资源状态节点 v_{i+n},所经过的节点为 v_i、v_{i+1}、…、v_{i+n},使得其任意 2 个相邻节点 v_{i+k}、v_{i+k},其中,$0\leqslant k\leqslant n-1$,总有 $t_{i+k\to i+k+1}\in T$,并且变迁序列是有限的,则称变迁序列 $<t_{i\to i+1}, t_{i+1\to i+2}, \cdots, t_{i+n-1\to i+n}>$ 为一条攻击路径,记为 *STEPT*,可用下式表示:

$$STEPT = \frac{\{v_i + n + 1 \in VG\} \otimes BNWAG}{t_i + k \to i + k + 1} \tag{1}$$

图 2 共有 3 条攻击路径即:

$STEPT_1 = <t_{0\wedge1\to3}, t_{3\to5}, t_{5\to8}, t_{8\to10}>$

$STEPT_2 = <t_{0\wedge1\to3}, t_{2\to4}, t_{3\wedge4\to7}, t_{7\to9}, t_{9\to10}>$

$STEPT_3 = <t_{2\to4}, t_{4\to6}, t_{6\to8}, t_{8\to10}>$

攻击者相继占有图 2 的所有节点后到达 v_{10},对此过程做出如下分析:

1)从理论上来说,只考虑各个节点的可达概率 $P(v_i)$,由于 $P(v_i)>0$,因此攻击者可以在 3 条攻击路径之中任意选择实施攻击,最终占有目标资源状态节点 v_{10}。

2)从实际情况来说,攻击成本在某种程度上影响攻击行为的发生概率,由于 $cost(t_{6\to8})$ 远大于 $cost(t_{5\to8})$,所以攻击者会选择发动攻击行为 $e_{5\to8}$,提高了攻击行为 $e_{5\to8}$ 发生的概率,降低了攻击行为 $e_{6\to8}$发生的概率。

3)若考虑节点之间的置信度,以节点 v_5、v_6、v_7 为例,由于节点 v_5、v_6、v_7 存在一定的相关性,导致 v_5、v_6、v_7 并非独立的影响 v_8、v_9,由于成本 $cost(t_{6\to8}) > cost(t_{5\to8})$,使得 $I(v_5)$ 和 $I(v_7)$ 随着攻击行为 $e_{6\to8}$ 发生概率的减小而增大。

2 基于可达概率和攻击成本的算法分析

在主观的网络攻击图中,节点可达概率以及攻击成本影响攻击路径的选择,在对攻击路径的分析中,往往会忽略作为智能主体的攻击者,优先选择花费时间较短的攻击路径发动攻击这一事实,造成攻击图中存在着大量可能性很低的攻击路径,对于攻击者而言,攻击行为时间越短,所付出的成本也就越小,节点的可达概率也就越大,成功的可能性也就越大,由于攻击路径随目标网络节点数目的增多呈指数增长,攻击路径的增多就必然导致攻击图规模增大,鉴于此,本文采用正向广度优先遍历搜索方法,从攻击者的角度模拟攻击路径,为避免攻击图规模过大,本文提出基于时间增益补偿率的攻击路径优化算法。

2.1 节点可达概率

为客观描述各个资源状态节点可达的不确定性,并结合贝叶斯定理,推断各状态节点的可达概率,即 $P(v_j)$的大小。$\forall v_j \in VG$,V_0为攻击的初始状态,一般 $P(V_0)=1$。

$\forall v_j \in VG$,根据贝叶斯定理可得:

$$P(v_j) = P(e_{i\to j})P(v_i) \tag{2}$$

其中,$v_j = Post(v_i)$。

式(3)是在不考虑攻击节点之间的关系得出来的资源状态节点的 v_i的可达概率,但在实际的网络攻击图中往往会出现节点之间 AND-OR 关系,假设 $\mathrm{Pre}(v_i)$ 相互独立,得出基于 AND 关系的节点可达概率 $P(v_j)$如式(3)所示和 OR 关系的节点可达概率 $P(v_j)$如下:

$$P(v_j) = \prod_{v_j = post(v_i)} P(e_{i\to j}) \times P(v_i) \tag{3}$$

$$P(v_j) = 1 - \prod_{v_j = post(v_i)} (1 - P(e_{i\to j}) \times P(v_i)) \tag{4}$$

假设图 2 中资源状态节点 v_0、v_1、v_2 的可达概率为 0.8、0.7、0.7,攻击行为 $v_1\rightarrow v_3$、$v_0\rightarrow v_3$、$v_3\rightarrow v_5$、$v_3\rightarrow v_7$、$v_2\rightarrow v_4$、$v_4\rightarrow v_6$、$v_4\rightarrow v_7$、$v_5\rightarrow v_8$、$v_6\rightarrow v_8$、$v_7\rightarrow v_9$、$v_8\rightarrow v_{10}$、$v_9\rightarrow v_{10}$ 成功的概率分别为 0.7、0.6、0.7、0.6、0.8、0.6、0.7、0.8、0.6、0.7、0.9、0.7,计算结果如表 1 所示。

表 1　各节点概率计算结果

节点	公式	概率
v_3	式(3)	0.23
v_4	式(2)	0.56
v_5	式(2)	0.16
v_6	式(2)	0.33
v_7	式(3)	0.05
v_8	式(4)	0.30
v_9	式(2)	0.03
v_{10}	式(4)	0.29

在对结果的分析中发现 $P(v_7)$ 的可达概率仅为 0.05,可用 3 种场景来解释:1)对时间不敏感的场景,该场景可以忽略时间的影响,操作成本占的权重居于主导,付出的操作成本很高。2)对时间非常敏感的场景,该场景可以忽略其他的影响因素,时间成本占的权重居于主导,付出的时间成本很高。3)综合考虑的场景,各种因素成本权重分布均匀,各种因素成本和很高。

2.2　攻击成本分析

攻击成本($cost(t_{i\rightarrow j})$)由操作成本、时间成本和风险成本构成。

资源状态节点的节点度越高,攻击者付出的操作成本越高,节点度与操作成本正相关,因此,可用节点度作为操作成本的一种度量,由此提出了基于节点度的操作成本的数学模型:

$$cost_{\text{operation}}(t_{i\rightarrow j})=\vartheta\times S(v_j) \tag{5}$$

其中,ϑ 表示相关性系数,用来量化操作成本和节点度之间的关联程度且 $\vartheta\in(0,1)$,$S(v_j)$ 表示节点 v_j 的节点度,接下来为方便起见用 $t_{i\rightarrow j}$ 表示资源状态节点之间的变迁关系。

除了操作成本之外,攻击者在实施攻击过程还要付出一定的时间成本,因此,再对子攻击路径成本分析时,还要考虑时间成本对攻击行为的影响。

时间系数(Γ)是用来衡量某次攻击所消耗时间的一种度量(即时间成本的度量),时间系数由两方面来决定,一方面取决于攻击目标自身影响系数($\theta(v_j)$),另一方面取决于攻击行为的影响系数($\sigma(e_{i\rightarrow j})$),因此,时间系数 Γ 的数学模型为:

$$\Gamma_{\text{time}}(t_{i\rightarrow j})=\sigma(e_{i\rightarrow j})\times\theta(v_j) \tag{6}$$

随着资源状态节点的节点度的增加,所付出的操作成本也随之增加,攻击者不可避免的要增加一系列必要的元操作,从而增加了时间成本,若攻击行为只包含一些简单的元操作,但要不断地重复这些元操作才能占有目标节点,目标节点的节点度低,但所花费的时间却很高,因此操作成本与时间成本并不是绝对的正比关系,时间成本是有元操作时间成本序列决定的。然而不同经验的攻击者花费的时间成本不同,依据以上分析,时间成本为:

$$cost_{\text{time}}(t_{i\rightarrow j})=\Gamma_{\text{time}}(t_{i\rightarrow j})\times\sum time(meta-operation)\times\pi(t_{i\rightarrow j}) \tag{7}$$

其中,$\pi(t_{i\rightarrow j})\in(0,1]$,$\Gamma_{\text{time}}(t_{i\rightarrow j})$ 表示时间系数,$time(meta-operation)$ 表示元操作时间成本,$\sum time(meta-operation)$ 表示元操作时间成本序列和,$\pi(t_{i\rightarrow j})$ 表示攻击者经验系数。

被攻击目标的攻击复杂度(本文用操作成本来表示)越高,攻击者花费的时间成本越多,攻击者暴露的风险就越大。依据以上分析,风险成本有如下公式:

$$cost_{\text{risk}}(t_{i\rightarrow j})=cost_{\text{operation}}(t_{i\rightarrow j})\times cost_{\text{time}}(t_{i\rightarrow j}) \tag{8}$$

其中,$cost_{\text{operation}}(t_{i\rightarrow j})$ 为操作成本,$cost_{\text{time}}(t_{i\rightarrow j})$ 为时间成本。

由式(8)可以看出,风险成本随着变量的变化而呈现出的变化趋势和实际攻击中的变化趋势是相符合的,从而得出攻击成本的计算式:

$$cost(t_{i\rightarrow j})=\tau cost_{\text{operation}}(t_{i\rightarrow j})+\varepsilon cost_{\text{time}}(t_{i\rightarrow j})+\mu cost_{\text{risk}}(t_{i\rightarrow j}) \tag{9}$$

其中,τ、ε、μ 分别为 3 种成本权重且 $\tau+\varepsilon+\mu=1$。

由式(8)可以看出,风险成本是由操作成本和时间成本决定的,因此,下面主要针对操作成本、时间成本进行分析。在以往对攻击成本的分析中,往往把操作成本作为路径存在冗余的判断条件,但没有考虑攻击过程中所花费的时间成本,若存在对实时性要求很高的场景,时间成本的考虑是非常必要的。在式(9)中,假设操作成本与风险成本所占的权重(即 τ 和 μ)很小,可以忽略不计,能够明显地看出攻击成本与时间成本正相关,然而这种假设在实际的网络攻击图是客观存在的。

2.3　节点可达概率与攻击成本关系分析

在网络攻击图中,所熟知的是攻击复杂度(本文用操作成本来表示)和节点可达概率成反比关系,但没有考虑时间成本的对节点可达概率的影响,因此,首先给出可达概率与攻击成本之间的关系式,其次对时间成本与可达概率之间的关系进行分析,关系式如下:

$$P(v_j)=f\left(\frac{1}{cost(t_{i\rightarrow j})}\right)\Rightarrow P(v_j)=f\left(\frac{1}{\tau cost_{\text{operation}}(t_{i\rightarrow j})+\varepsilon cost_{\text{time}}(t_{i\rightarrow j})+\mu cost_{\text{risk}}(t_{i\rightarrow j})}\right) \tag{10}$$

其中,$P(v_j)$ 表示 v_j 的可达概率,$cost(t_{i\rightarrow j})$ 表示攻击成本,$f(x)$ 表示关系函数。

在以往的场景中,由于操作成本与节点可达概率成反比关系,但这种关系仅仅适用于对时间不敏感的场景中,倘若存在对时间非常敏感的场景,攻击者会更加看重时间成本,如随着时间的推移,网络防御系统或网络管理员可能发现攻击者部分或全部的攻击行为,采用断开物理连接或者关闭服务等方式阻止攻击者的下一步攻击。

在式(10)中,假设操作成本与风险成本所占的权重(τ 和 μ)很小,忽略不计,可以看出,时间成本与节点可达概率负相关。因此,可以得出,时间成本影响攻击行为的发生概率。

2.4 时间增益补偿率

在现实的网络环境中,攻击者遵循的是收益最大化原则,对实时性的要求也很高,若攻击行为发生的时间持续较长,目标资源状态节点的可达概率随时间的增长急速下降,同时其攻击成本上升很快,攻击者的攻击意愿就会下降很快,若时间成本(h)达到了指定的阈值(m),攻击者会放弃攻击,根据以上分析,时间增益补偿率有如下公式:

$$\Delta=\frac{h-m}{m}=\frac{cost_{\text{time}}(t_{i\to j})-m}{m} \tag{11}$$

其中,时间增益补偿率依附于有向边上。可以看出,当 $\Delta\leqslant 0$ 时,攻击者认为发动攻击是可行的,若时间增益补偿率 Δ 正向越趋近于 0,目标资源状态的可达概率也就越小,只有当 $\Delta>0$ 时,攻击者付出的时间成本超出接受的范围,就放弃攻击。

2.5 攻击路径生成算法描述

定义 4(变迁关系集) $\forall v_i\in V,\exists v_j=Post(v_i)$,若 v_i、v_j之间存在一条 $<v_i,v_j>$ 的有向边 $e_i\to j$,则称 v_i、v_j之间存在变迁关系,用 $t_i\to j$ 来表示。由变迁关系组成的集合,称为变迁关系集(Transiton Relation Set),记为 *TRS*。

由 *STEPT* 和 *TRS* 可以看出,两者都是由 $t_{i\to j}$作为基本元素组成的集合,因此,当 *TRS* 包含网络攻击图所有变迁关系时,遍历 *TRS* 就可以得到所有 *STEPT* 集合,因此有 *STEPTTRS*。

为了形象地展示攻击路径的生成过程,用⊙表示资源状态变迁 $t_{i\to j}$或攻击行为 $e_{i\to j}$发生,用◎表示不可能发生,其中,默认情况 $t_{i\to j}\leftarrow\odot$,本文根据上文的数学模型,并结合专家经验为每条有向边 $e_{i\to j}$赋权值 w;以根节点 v_0为起始节点,逐步移除以其为起始节点的有向边 $e_{i\to j}$,并将变迁关系 $t_{i\to j}$加入到集合 TRS_k中;以此类推,依照移除有向边 $e_{i\to j}$的某一拓扑顺序 ψ,逐步得到所有节点之间的变迁关系集 *TRS*,为每一条有向边 $e_{i\to j}$赋予权值 w 后得到的攻击图如图 3 所示。

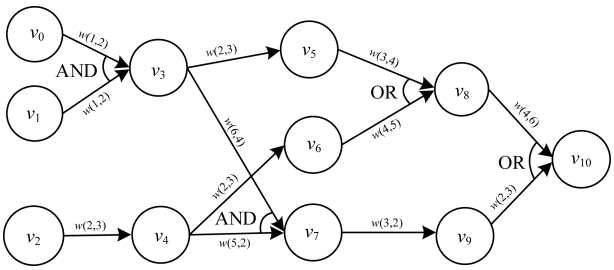


图 3 变迁关系生成图

假设拓扑序 $\psi=\{e_{0\to3},e_{1\to3},e_{2\to4},e_{3\to5},e_{3\to7},e_{4\to7},e_{4\to6},e_{5\to8},e_{6\to8},e_{7\to9},e_{8\to10},e_{9\to10}\}$,具体实现过程如下:

1)移除 $e_{0\to3}$,$\Delta=\dfrac{1-2}{2}<0$,$e_{0\to3}\leftarrow\odot$。

2)移除 $e_{1\to3}$,$\Delta=\dfrac{1-2}{2}<0$,$e_{1\to3}\leftarrow\odot$,$TRS_0=\{t_0\wedge 1\to2\}$。

3)移除 $e_{2\to4}$,$\Delta=\dfrac{2-3}{3}<0$,$e_{2\to4}\leftarrow\odot$,$TRS_1=TRS_0\cup\{t_{2\to4}\}$。

4)移除 $e_{3\to5}$,$\Delta=\dfrac{2-3}{3}<0$,$e_{3\to5}\leftarrow\odot$,$TRS_2=TRS_1\cup\{t_{3\to5}\}$。

5)移除 $e_{3\to7}$,$\Delta=\dfrac{6-4}{4}>0$,$e_{3\to7}\leftarrow◎$。

6)移除 $e_{4\to7}$,$\Delta=\dfrac{5-2}{2}>0$,$e_{4\to7}\leftarrow◎$,$t_{3\wedge4\to7}\leftarrow◎$,$TRS_3=TRS_2\cup\{t_{3\wedge4\to7}\}$。

7)移除 $e_{4\to6}$,$\Delta=\dfrac{2-3}{3}<0$,$e_{4\to6}\leftarrow\odot$,$TRS_4=TRS_3\cup\{t_{4\to6}\}$。

8)移除 $e_{5\to8}$,$\Delta=\dfrac{3-4}{4}<0$,$e_{5\to8}\leftarrow\odot$,$TRS_5=TRS_4\cup\{t_{5\to8}\}$。

9)移除 $e_{6\to8}$,$\Delta=\dfrac{4-5}{5}<0$,$e_{6\to8}\leftarrow\odot$,$TRS_6=TRS_5\cup\{t_{6\to8}\}$。

10)移除 $e_{7\to9}$,$\Delta=\dfrac{3-2}{2}>0$,$e_{7\to9}\leftarrow◎$,$t_{7\to9}\leftarrow◎$,$TRS_7=TRS_6\cup\{t_{7\to9}\}$。

11)移除 $e_{8\to10}$,$\Delta=\dfrac{4-6}{6}<0$,$e_{8\to10}\leftarrow\odot$,$TRS_8=TRS_7\cup\{t_{8\to10}\}$。

12)移除 $e_{9\to10}$,$\Delta=\dfrac{2-3}{3}<0$,$e_{9\to10}\leftarrow\odot$,$TRS_9=TRS_8\cup\{t_{9\to10}\}$。

13)遍历 TRS_9得到的攻击路径:

$STEPT_1 = < t_{0\wedge 1\to 3}, t_{3\to 5}, t_{5\to 8}, t_{8\to 10} >$

$STEPT_2 = < t_{0\wedge 1\to 3}, t_{2\to 4}, t_{3\wedge 4\to 7}, t_{7\to 9}, t_{9\to 10} >$

$STEPT_3 = < t_{2\to 4}, t_{4\to 6}, t_{6\to 8}, t_{8\to 10} >$

14)遍历 TRS_9 查找所有标志为◎的 $t_{i\to j}$,即 $t_{3\wedge 4\to 7}$、$t_{7\to 9}$。

15)输出可能的攻击路径:

$STEPT_1 = < t_{0\wedge 1\to 3}, t_{3\to 5}, t_{5\to 8}, t_{8\to 10} >$

$STEPT_3 = < t_{2\to 4}, t_{4\to 6}, t_{6\to 8}, t_{8\to 10} >$

在步骤 13 中得出的 $STEPT_1$、$STEPT_2$、$STEPT_3$ 与前面得到的 3 条攻击路径相比,通过变迁关系得到的攻击路径与实际攻击路径保持一致,降低了攻击路径生成的复杂度,由于 $cost_{\text{time}}(t_{3\wedge 4\to 7})$ 所付出的时间成本很高,因此导致 $P(v_7)$ 的概率很小,所以,攻击者放弃攻击,与前面的分析保持一致,从而证明本文方法的有效性。

3　改进贝叶斯似然加权算法

算法 1 目的是消去冗余攻击路径。首先,寻找有向边 $e_{i\to j}$,并对 $e_{i\to j}$ 进行时间增益补偿率 Δ 判断:$\Delta(e_{i\to j})\leqslant 0$,$e_{i\to j}\leftarrow\odot$,否则 $e_{i\to j}\leftarrow$◎,$t_{i\to j}\leftarrow$◎,接下来由变迁关系集 *TRS* 得到变迁关系集合 *STEPT*,并去掉标记◎的 $t_i\to j$ 所在的攻击路径 $STEPT_j$,这些攻击路径即为冗余路径,最后,输出变迁关系集合 $STEPT_i$。

算法 1　变迁关系 *STEPT*(*BNWA*,$STEPT_i$)

输入　网络攻击图 *BNWAG*,权重 *w*,变迁关系集合 $STEPT_i$,有向边集合 *E*,变迁关系集 *TRS*,逻辑关系集合 *D*,*BNWAG* 节点拓扑序 ψ,任意节点 M_i、M_j

输出　网络攻击图的变迁关系节点集合

1. ψ←BNWAG,TRS←φ,$STEPT_i$←φ
2. FOR(根据节点拓扑序 ψ 查找每一个节点变量 M_i)
3. 在 BNWAG 中找出与节点 M_i 存在变迁关系的节点 M_j
4. IF($e_{i\to j}\in$(E)
5. IF(时间增益补偿率 $\Delta(e_{i\to j})\leqslant 0$
6. $e_{i\to j}\leftarrow\odot$
7. ELSE
8. $e_{i\to j}\leftarrow$◎,$t_{i\to j}\leftarrow$◎
9. END IF
10. IF(射入 M_j 的边之间存在 AND 关系)
11. $TRS_{i+1}\leftarrow TRS_i\cup\{t_{i\wedge i+1\wedge\ldots\wedge i+n\to j}\}$
12. ELSE
13. $TRS_{i+1}\leftarrow TRS_i\cup\{t_{i\to j}\}$ $TRS_{i+1}\leftarrow TRS_i\cup\{t_{i\to j}\}$
14. END IF
15. END IF
16. END FOR
17. STEPT⊆TRS
18. 遍历集合 TRS,查找所有标志为◎的 $t_{i\to j}$
19. IF $STEPT_i$ 不为空集
20. RETURN $STEPT_i$

由于似然加权算法的随机性,传统似然加权算法在解决概率为 0 和 1 的节点的抽样问题时存在一定的缺陷,因此能够在传统的基础上对其改进,以网络攻击图节点 v_j 为例,若攻击者放弃攻击节点 v_j,节点 v_j 的置信度 $I(v_j)=0$,由于抽样的不确定性,导致 $I(v_j)>0$,影响节点置信度计算的准确性,若攻击者选择攻击 v_j 时(其中,v_j 为关键节点,即到达目标节点的必经节点),节点置信度 $I(v_j)=1$,由于抽样的不确定性,$I(v_j)<1$,影响节点置信度计算的准确性,因此在抽样前,首先赋 *X* 的取值为 False,*Y* 的取值为 True,然后再进行抽样,这样保证了抽样过程总是按照相应的概率分布进行,改进后如算法 2 所示。

算法 2　改进似然加权算法 *Likelihoodweighting*(*BNWAG*,*m*,*E*,*e*,*Q*,*q*,ρ)

输入　网络攻击图 *BNWAG*,有效样本量 *m*,证据变量集合 *T*,证据变量取值 *t*,查询变量集合 *Q*,查询变量取值 *q*,节点拓扑序 ρ

输出　对 $P(Q=q|E=e)$ 的近似值

1. ρ←BNWAG
2. $\omega_e\leftarrow 0$,$\omega_{q,e}\leftarrow 0$
3. For(从 1 ~ m)
4. $D_i\leftarrow\varphi$
5. For(遍历 ρ 中的每一个节点)
6. $T_y\leftarrow$AbandonNodeSet(TRS,TEMP,Q,I_Q)
7. $T_x\leftarrow$KeyNodeSet(TRS,$STEPT_i$,TEMP,Q,I_Q)
8. IF($X\in T_x$)
9. x←False
10. ELSE IF($X\in T_y$)
11. x←True
12. ELSE
13. IF($x\in T$) then
14. x←X 的观测值
15. ELSE
16. x←从 P(X|Pre(X))获得的结果
17. END IF
18. END IF
19. END FOR
20. $D_i\leftarrow D_i\cup\{X=x\}$
21. $\omega_i\leftarrow\prod_{x\in E}P(X|I(X))|D_i$
22. $\omega_e\leftarrow\omega_e+\omega_i$
23. IF($q\in D_i$) then
24. $\omega_{q,e}\leftarrow\omega_{q,e}+\omega_i$
25. END IF
26. END WHILE
27. RETURN$\omega_{q,e}/\omega_e$

通过算法 2 可以得到节点的置信度分布。首先,根据攻击图 *BNWAG* 得到节点拓扑序 ρ,第 4 行~第 20 行为样本的生成过程,第 6 行和第 8 行生成并保证置信度为 0 的样本的取值为

False,第 7 行、第 11 行生成并保证置信度为 1 的样本的取值为 True,第 14 行 X 是根节点,抽样结果作为其观测值,第 16 行为为非根节点,按照概率分布 $P(X|\mathrm{Pre}(X))$ 进行抽样。这样保证了抽样过程总是按照相应的概率分布进行。

算法 3 是为了得到关键节点,首先集合 *TEMP* 中所有变迁关系状态进行判断,若 Q 的状态为◎且 Q 属于某一个变迁关系集合 $STEPT_i$,那么这个变迁关系集合为干扰集合,直接舍弃。保留有效的集合,最后得到所有的关键节点集合。

算法 3 关键节点集 *KeyNodeSet*(*TRS*, $STEPT_i$, *TEMP*, Q, I_Q)

输入 *TRS*←变迁关系集,$STEPT_i$ 变迁关系集合,*TEMP*←$t_{i\wedge i+1,\cdots,i+n\to j}$ 集合,Q←*TEMP* 变迁关系集合中的任意变量,Y←Q 变迁关系对应的目标资源状态节点变量,I_Q←Q 状态

输出 V_y←关键节点变量集合

1. V_y←φ, TRS_k←φ
2. FOR(TEMP 中每一个变量 Q)
3. IF(I_Q = ◎ ∧ Q 变迁结果 Y)
4. IF Q ∈ $STEPT_i$
5. AbandonChangeSet($STEPT_i$)
6. ELSE
7. TRS_K←TRS_K ∪ $STEPT_i$
8. END IF
9. END FOR
10. V_y←集合 TRS_K 逐项同或获得的 Y
11. RETURN V_y

算法 4 是为了得到攻击者放弃攻击的节点。首先遍历变迁关系集合,若变迁关系状态标志为◎,则认为该变迁结果的节点 Y 为攻击者放弃攻击的节点,将节点 Y 放到集合 V_N 中,否则舍弃该节点。

算法 4 放弃节点集 *AbandonNodeSet*(*TRS*, *TEMP*, Q, I_Q)

输入 *TRS*←变迁关系集,*TEMP*←$t_{i\wedge i+1,\cdots,i+n\to j}$ 集合,Q←*TEMP* 变迁关系集合中的任意变量,Y←Q 变迁关系对应的目标资源状态节点变量,I_Q←Q 状态

输出 V_N 关键节点变量集合

1. V_N←φ
2. FOR(TEMP 中每一个变量 Q)
3. IF(Q 变迁结果 Y∧I_Q = ◎)
4. V_N←V_N ∪ Y
5. ELSE
6. AbandonTransitonRelation(Q)
7. END IF
8. END FOR
9. RETURN V_N

4 算法分析及验证

4.1 改进贝叶斯似然加权算法

实验采用 C++编程实现传统似然加权算法和改进似然加权算法。表 2 是由图 2 采用传统似然加权算法计算的节点置信度,表 3 是由图 3 采用改进贝叶斯加权算法计算的节点置信度,有效实验样本 4 000个,a 表示 $\theta_1=0.4$、$\theta_2=0.6$ 情况下条件概率,b 表示 $\theta_1=0.6$, $\theta_2=0.4$ 情况下条件概率。

表 2 传统似然加权推理结果

节点置信度	条件概率 a	条件概率 b
$I(v_0)$	1.00	1.00
$I(v_1)$	1.00	1.00
$I(v_2)$	1.00	1.00
$I(v_3)$	0.18	0.26
$I(v_4)$	0.24	0.30
$I(v_5)$	0.26	0.24
$I(v_6)$	0.38	0.36
$I(v_7)$	0.46	0.44
$I(v_8)$	0.49	0.47
$I(v_9)$	0.36	0.36
$I(v_{10})$	0.35	0.30

表 3 改进似然加权推理结果

节点置信度	条件概率 a	条件概率 b
$I(v_0)$	1.00	1.00
$I(v_1)$	1.00	1.00
$I(v_2)$	1.00	1.00
$I(v_3)$	0.25	0.32
$I(v_4)$	0.33	0.36
$I(v_5)$	0.45	0.43
$I(v_6)$	0.52	0.51
$I(v_7)$	0.00	0.00
$I(v_8)$	1.00	1.00
$I(v_9)$	0.00	0.00
$I(v_{10})$	1.00	1.00

为了更直观地观察算法改进前后节点置信度的变化,以 $STEPT_2$ 和 $STEPT_3$ 为例,绘制出如图 4 和图 5所示的节点置信度变化图,用 *traditional* 表示传统似然加权,*improve* 表示改进的似然加权,比如 *improve*(b)表示条件概率为 b 时,改进似然加权算法的节点置信度变化。

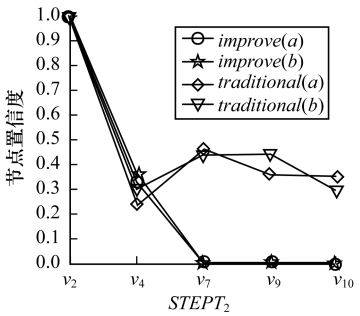


图 4　$STEPT_2$ 节点置信度对比

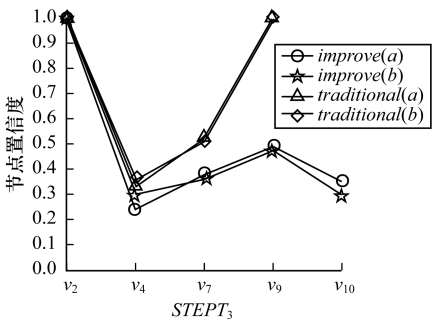


图 5　$STEPT_3$ 节点置信度对比

由图 4 可以看出,由于传统似然加权算法没有考虑节点置信度为 0 的情况 ,因此不能消除冗余路径,然而改进似然加权算法考虑到了这点,即当攻击者认为攻击花费的时间成本大于某一阈值时,就会放弃攻击,那么该节点的置信度就置为 0,即节点 v_7、v_9 置信度 $I(v_7)$、$I(v_9)$ 都是 0,能够有效地消除路径冗余。在图 5 中由于 $STEPT_3$ 为冗余路径,但传统似然加权算法没有考虑冗余路径,导致节点的置信度偏低,而改进的似然加权算法考虑到了这点,因而图 5改进算法的节点置信度比传统算法要高,同时改进算法也考虑了关键点的置信度,即攻击者为获取目标资源必须占有的节点即为关键点,关键点的置信度为 1,即 $I(v_8)$、$I(v_{10})$ 都为 1。

4.2　攻击路径优化

在图 3 中,以 $P(v_{10}\mid v_0,v_1,v_2)$ 作为所求概率值 $P(v_7=\text{True}\mid v_0=\text{True},v_1=\text{True},v_2=\text{True})$ 的抽样统计项,采用 12 个采样时间点分别对 $STEPT_1$、$STEPT_2$、$STEPT_3$ 路径进行抽样分析。用 φ_1 表示 $STEPT_1$ 的抽样概率,φ_2 表示 $STEPT_2$ 的抽样概率,φ_3 表示 $STEPT_3$ 的抽样概率,用 t_i 表示采样时间,i 值越大表示采样时间越长。抽样结果如表 4 所示。由实验结果分析可知,采样时间的不同,攻击者对每条攻击路径的攻击概率也不相同,为了更加直观地观察攻击路径的预测效果,绘制如图 6 所示的结果。

表 4　攻击路径生成的推理结果

抽样时间/h	抽样概率 φ_1	抽样概率 φ_2	抽样概率 φ_3
t_1	0.30	0.22	0.20
t_2	0.32	0.29	0.26
t_3	0.40	0.38	0.36
t_4	0.50	0.47	0.43
t_5	0.55	0.52	0.51
t_6	0.60	0.59	0.63
t_7	0.64	0.66	0.69
t_8	0.70	0.69	0.72
t_9	0.73	0.00	0.75
t_{10}	0.78	0.00	0.77
t_{11}	0.82	0.00	0.80
t_{12}	0.85	0.00	0.83

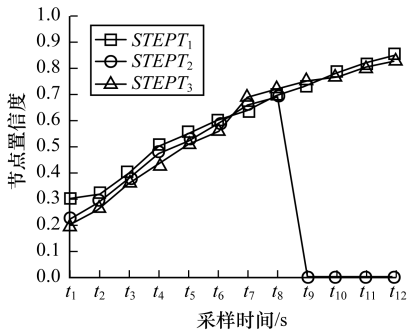


图 6　攻击路径优化结果

如图 6 所示,在攻击开始时,随着攻击进度的推进,攻击者掌握的经验和资源不断增加,攻击发生的概率 Θ_1 以及攻击成功的概率 Θ_2 也在不断地增加,因此,$STEPT_1$、$STEPT_2$、$STEPT_3$ 发生的概率呈现上升的趋势,由于攻击时间不断增长,攻击行为被 IDS 或者管理员发现的概率也在不断地增加,当达到一定的临界值时,攻击行为被发现,意味着条件概率 Θ_1 和 Θ_2 降为 0。结果如 $STEPT_2$ 所示。由此可以看出,该模型能够为检测方提供有效而量化的攻击概率,便于管理决策人员作出正确的决策。

5　结束语

本文引入时间成本的概念,提出一种基于时间增益补偿率攻击路径优化算法,从而尽可能消除冗余节点,并在此基础上,改进贝叶斯似然加权算法,以提高节点置信度计算的准确性,并通过对比结果分析,证明了该算法的有效性。然而,本文对时间成本的取值依赖于专家经验,因此,如何获取非专家经验的时间成本,将是下一步的研究内容。

(下转第 198 页)

参考文献

[1] 彭智勇,彭煜玮. PostgreSQL 数据库内核分析[M]. 北京:机械工业出版社,2012.

[2] MOMJIAN B. PostgreSQL introduction and concepts[J]. Journal of Conflict Resolution, 2001 (3): 353-355.

[3] WONG C K, GOUDA M, LAM S S. Secure group communications using key graphs[C]//Proceedings of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. New York, USA: ACM Press, 1998: 68-79.

[4] SHERMAN A T, MCGREW D A. Key establishment in large dynamic groups using one-way function trees[J]. IEEE Transactions on Software Engineering, 2003, 29(5): 444-458.

[5] HAN Yiliang, GUI Xiaolin, WU Xuguang, et al. Proxy encryption based secure multicast in wireless mesh networks[J]. Journal of Network and Computer Applications, 2010, 34(2): 469-477.

[6] PERRIG A, SONG D, TYGAR J D. ELK, a new protocol for efficient large-group key distribution[C]// Proceedings of IEEE Symposium on Security and Privacy. Washington D. C., USA: IEEE Press, 2001: 247-262.

[7] MITTRA S. A framework for scalable secure multicasting[C]//Proceedings of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. New York, USA: ACM Press, 1997: 277-288.

[8] SAHAI A, WATERS B. Fuzzy identity-based encryption[C]//Proceedings of International Conference on Theory and Applications of Cryptographic Techniques. Berlin, Germany: Springer-Verlag, 2005: 457-473.

[9] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]//Proceedings of ACM Conference on Computer and Communications Security. New York, USA: ACM Press, 2006: 89-98.

[10] 李春梅,杨小东,周思安,等. 一种面向社交网络的细粒度密文访问控制方案[J]. 计算机工程,2015, 41(2): 117-121.

[11] RIVEST R L. All-or-nothing encryption and the package transform[C]//Proceedings of International Workshop on FAST Software Encryption. Washington D. C., USA: IEEE Press, 1997: 210-218.

[12] 周德华. 代理重加密体制的研究[D]. 上海:上海交通大学,2013.

[13] 王会歌,王彩芬,曹 浩,等. 具有代理重加密性质的自认证签密方案[J]. 计算机工程,2012,38(13): 122-124.

[14] 郑志恒,张敏情,戴晓明,等. 高效的基于代理重加密的云存储访问控制方案[J]. 电子技术应用,2016, 42(11): 99-101.

[15] SYALIM A, NISHIDE T, SAKURAI K. Realizing proxy re-encryption in the symmetric world[J]. Communications in Computer & Information Science, 2011, 251: 259-274.

编辑 刘 冰

(上接第 191 页)

参考文献

[1] 2016 年中国互联网安全报告[EB/OL]. [2016-11-21]. http://www.cert.org.cn/publish/main/46/index.html.

[2] 高 妮,高 岭,贺毅岳,等. 基于贝叶斯攻击图的动态安全风险评估模型[J]. 四川大学学报(工程科学版),2016,48(1): 111-118.

[3] 尚大鹏,张 冰,周 渊,等. 一种深度优先的攻击图生成方法[J]. 吉林大学学报(工学版),2009,39(2): 446-452.

[4] GHASEMIGOL M, GHAEMI-BAFGHI A, TAKABI H. A comprehensive approach for network attack forecasting[J]. Computers and Security, 2016, 58(1): 83-105.

[5] 彭 武,胡昌振,姚淑萍. 基于攻击路径图的入侵意图识别[J]. 北京理工大学学报,2010,30(9): 1077-1081.

[6] KAYNAR K. A taxonomy for attack graph and usage in network security[J]. Information Security and Applications, 2016, 29(1): 27-56.

[7] LV Huiying, ZHANG Yuan, WANG Jie. Network threat identification and analysis based on a state transition graph[J]. Cybernetics & Information Technologies, 2013(13): 51-61.

[8] POOLSAPPASIT N, DEWRI R, RAY I. Dynamic security risk management using bayesian attack graphic[J]. IEEE Transactions on Dependable and Secure Computing, 2012, 9(1): 61-74.

[9] QIU Hui, WANG Kun. Real-time network attack intention recognition algorithm[J]. International Journal of Security and Its Applications, 2016, 10(4): 51-62.

[10] CHANDRAN R, YAN W. Attack graph analysis for network anti-forensics[J]. International Journal of Digital Crime & Forensics, 2014, 6(1): 28-50.

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

[12] YE Yun, XU Xishan, JIA Yan, et al. An attack graph-based probabilistic computing approach of network security[J]. Chinese Journal of Computers, 2010, 33(10): 1987-1996.

[13] 贺志强,楼 芳,李 亮. 基于攻击距离的攻击图优化方法[J]. 计算机工程与科学,2012,34(2): 9-12.

[14] 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.

[15] 吴泓润,覃 俊,郑波尽. 基于代价的复杂网络抗攻击性研究[J]. 计算机科学,2012,39(8): 224-227.

编辑 刘 冰