

面向大规模工控网络的关键路径分析方法

张耀方^{1,2}, 张哲宇³, 曲海阔^{1,2}, 张格³, 王子博^{1,2}, 王佰玲^{1,2}

(1. 哈尔滨工业大学(威海)计算机科学与技术学院, 山东 威海 264209;

2. 国家工业信息安全发展研究中心, 北京 100040;

3. 哈尔滨工业大学网络空间安全研究院, 黑龙江 哈尔滨 150006)

摘 要: 针对大规模工控网络攻击图的量化计算耗时高、消耗资源大的问题, 提出了一种大规模工控网络的关键路径分析方法。首先利用割集思想结合工控网络中的原子攻击收益, 计算贝叶斯攻击图关键节点集合, 解决目前割集算法只考虑图结构中节点关键性的问题。其次, 提出一种只更新关键节点攻击概率的贝叶斯攻击图动态更新策略, 高效计算全图攻击概率, 分析攻击图关键路径。实验结果表明, 所提方法在大规模工控攻击图的计算中, 不仅可以保证计算结果的可靠性, 而且能够大幅度降低方法耗时, 显著提升计算效率。

关键词: 关键节点; 关键路径; 攻击图; 贝叶斯网络; 工控网络

中图分类号: TP393

文献标识码: A

DOI: 10.11959/j.issn.2096-109x.2021069

Key path analysis method for large-scale industrial control network

ZHANG Yaofang^{1,2}, ZHANG Zheyu³, QU Haikuo^{1,2}, ZHANG Ge³, WANG Zibo^{1,2}, WANG Bailing^{1,2}

1. School of Computer Science and Technology, Harbin Institute of Technology (Weihai), Weihai 264209, China

2. China Industrial Control Systems Cyber Emergency Response Team, Beijing 100040, China

3. Research Institute of CyberSpace Security, Harbin Institute of Technology, Harbin 150006, China

Abstract: In order to solve the problem of high time-consuming and resource-consuming quantitative calculation of large-scale industrial control network attack graphs, a key path analysis method for large-scale industrial control networks was proposed. Firstly, the idea of cut set was used to calculate the key nodes set of Bayesian attack graph by combining the atomic attack income in industrial control network, which solved the problem that the current cut set algorithm only considers the key nodes in graph structure. Secondly, a dynamic updating strategy of Bayesian attack graph which only updated the attack probability of key nodes was proposed to efficiently calculate the attack probability of the whole graph and analyze the key path of attack graph. The experimental results show that the proposed method can not only ensure the reliability of the calculation results of large-scale industrial control attack

收稿日期: 2021-02-22; 修回日期: 2021-05-10

通信作者: 王佰玲, wbl@hit.edu.cn

基金项目: 国防基础科研计划(JCKY2019608B001)

Foundation Item: The National Defense Basic Scientific Research Program (JCKY2019608B001)

论文引用格式: 张耀方, 张哲宇, 曲海阔, 等. 面向大规模工控网络的关键路径分析方法[J]. 网络与信息安全学报, 2021, 7(6): 31-43.

ZHANG Y F, ZHANG Z Y, QU H K, et al. Key path analysis method for large-scale industrial control network[J]. Chinese Journal of Network and Information Security, 2021, 7(6): 31-43.

graphs, but also can significantly reduce the time consumption and have a significant improvement in the calculation efficiency.

Keywords: key node, key path, attack graph, Bayesian network, industrial control network

1 引言

随着传统工业网络与互联网的融合发展,工业网络被攻击的风险大大增加。尤其在工业控制(下文简称为工控)系统中,各种潜在风险导致系统安全性难以维护,造成巨大财产损失,甚至危害生命。近 10 年,工控事件频发,如 Stuxnet 事件^[1]、乌克兰电网事件^[2]以及 WannaCry 勒索病毒事件^[3]。越来越多的工控攻击事件引起人们的关注^[4],提高大众对工控网络安全整体认知能够对安全防御起到积极作用,因此对工控网络的风险分析迫在眉睫。

由于工控攻击呈现多步攻击趋势,常见的基于模糊测试、层次分析法、博弈论思想的风险评估方法无法全面分析多步攻击^[5]。Swiler 等^[6]提出攻击图这一概念,并将攻击图应用于网络威胁分析中。攻击图通过枚举可能攻击路径来预判攻击者对目标网络的攻击过程,定性地指导防御方加强网络中脆弱节点的防御。同时,将攻击图结合定量计算,可精准分析薄弱节点,以对其采取针对性防御措施,提高网络安全性^[7]。因此,对攻击图的量化计算及安全分析成为研究热点。

攻击图中关键路径的量化分析,是攻击者与防御者共同关心的问题。关键路径指在攻击图中对于成功到达某一节点完成攻击目标的概率最大的路径。受工控网络操作流程带来的级联效应的影响,满足工控网络的安全性需求要求攻击图中关键路径的分析有着更高的效率。同时,关键路径分析速度的提升意味着防御的提前。在实际工控环境中,底层数据采集、中枢逻辑控制以及顶层系统管理等工控网络的多种功能要求,导致攻击图中节点数较多且关系复杂。尤其是包含更多信息的动态更新的大规模攻击图,其量化计算中存在耗时长、占用资源多、动态更新效率低的问题。

在关键攻击路径的分析中,其计算效率取决于路径中节点概率、更新概率、累计概率的计算

效率。而对于关键节点的分析可以在动态更新时提前削减更新节点数量。目前对于节点关键性的量化多从图论方面考虑,但在工控环境中,节点关键性还受多种环境因素影响,不同漏洞节点的利用价值均不相等,因此还需结合工控环境因素对节点关键性进行度量。

针对以上问题,本文提出了一种利用计算贝叶斯攻击图关键节点集合进行部分节点概率更新,以计算关键攻击路径的方法。该方法考虑多种工控环境影响因素计算攻击图关键节点割集,并对割集中节点的概率进行更新计算,量化当前攻击图路径的被攻击概率。本文的主要贡献有以下两方面。

1) 在考虑图结构的节点关键性的基础上,为节点附加攻击收益指标,提出一种计算攻击图关键节点集合的算法。

2) 提出针对攻击图中关键节点进行攻击概率动态更新的策略,并在不同规模的工控网络下进行验证。

实验结果表明本文的方法能够高效计算工控网络攻击图的关键路径攻击概率,并可用于大规模工控网络的量化计算,为攻击图实时分析提供依据,为工控网络提前防御策略的制定提供直观的数据支撑。

2 相关研究

攻击图以图的形式将目标网络中的原子攻击关联起来,展示多步骤攻击路径,是进行定性安全分析的有效技术。攻击图安全分析领域中的研究热点包括基于攻击图的量化计算,将攻击图与贝叶斯网络等技术结合,在定性分析的基础上实现图节点以及攻击路径定量的准确分析。

在攻击图量化计算方面,Poolasappasit 等^[8]使用贝叶斯攻击图(BAG)量化网络遭到不同等级损害的风险。BAG 模型考虑常用的网络状态之间因果关系及其被利用的可能性,采用贝叶斯信念网络的概念,对不同安全条件下的损害进行评估。此模型适用于对环境动态分析的网络部署阶段,

在单目标分析的基础上,增加了多目标优化的决策权衡,但该方法的可扩展性和效率仍需提高。Gonzalez 等^[9]针对贝叶斯攻击图的静态和动态分析提出了一种有效的精确推断技术。经大量实验评估,该方法显示出在时间复杂度和空间复杂度方面的优势。杨英杰等^[10]对于属性攻击图模型中的转移概率度量问题,给出了单步威胁转移概率和综合威胁转移概率的度量算法,并解决了攻击图传递环路问题。

王辉等^[11]在攻击路径预测的研究中,改进了基于贝叶斯推理的似然加权法,并对子路径进行成本收益分析。该方法给出了攻击收益的数学模型,但其参数只能凭专家经验获得。张少俊等^[12]同样对似然加权法进行改进,并提出一种支持攻击证据之间时间偏序关系的攻击图节点置信度计算方法。该方法可有效提高置信度的准确性,同时算法复杂度低,适用于处理大规模攻击图量化分析问题。其与陈小军等^[13]提出的思想不同之处在于在攻击图模型中引入转移概率表来刻画单步攻击的不确定性,利用节点置信度推断攻击概率,计算最大攻击概率路径。该方法在小规模网络攻击图中可有效减少不可信报警,但由于其模型构建依赖专家知识库,不适用于大规模网络攻击图的意图推断和最大概率路径计算。

在量化计算指标方面,黄家辉等^[14]对工控系统脆弱性量化进行研究,考虑攻防强度、物理损失、信息损失因素,对漏洞等级进行划分。量化攻击过程中原子攻击的脆弱性,得到总攻击期望最大的路径。罗志勇等^[15]在计算原子攻击概率时,结合漏洞价值、攻击成本以及攻击收益等因素,对贝叶斯攻击图进行量化,并引入入侵意图更新评估模型,实现攻击图风险的动态评估。

但目前关于攻击图量化计算的研究中,大部分指标仍依靠专家经验获得。马春光等^[16]针对攻击图评估过于依赖专家知识库,无法考虑攻击者意愿的问题,提出了一个动态评估模型。其将攻击者意愿与原子攻击性质融入贝叶斯攻击图的动态推理中,提升了风险评估中攻击估算的合理性,适用于实际网络环境下的动态风险评估。高梦州等^[17]建立了基于专家知识库、脆弱性库的脆弱性利用规则库。利用参数的初级量化和矩阵判断法

计算攻击收益,对工控网络安全风险进行评估。

叶云等^[18]对于大规模攻击图的节点概率计算的耗时问题,提出了节点概率的近似计算方法,以牺牲概率精确度的策略提升计算效率,该方法适用于复杂的大规模攻击图。本文方法与其提升计算效率的目的相同,但叶云等的方法是针对攻击图全局节点进行近似计算,而本文针对耗时问题的优化思想在于利用节点概率计算方法,只针对部分节点概率进行动态更新。

在攻击图割集计算方面,刘贞宇等^[19]基于攻击图的几何性质,在保留攻击图基本网络结构的基础上对其进行分割,将大型攻击图划分成多个子攻击图进行分析。吴金字等^[20]将最优原子攻击修复集问题归结于攻击图中的最小 S-T 割集问题。利用原子攻击拆分加权重构攻击图,并采用最大流算法,对攻击图的最优原子集进行求解。其利用最优集思想解决问题的思路与本文类似,该方法计算效率高、可扩展性较好,但该方法仅对攻击图进行最优节点集合层面的分析,没有进行量化计算。彭武等^[21]利用最小割集思想对入侵意图进行阻断。该方法计算可达到攻击意图的所有攻击路径及其攻击概率,结合最小顶点割集思想,实现防护最小数目节点来达到阻断攻击意图的目的。与彭武等计算出路径概率,再利用最小割算法进行防御的思想相反,本文利用关键节点割集对攻击路径概率进行计算以实现风险量化。并且,彭武等提出的方法中最小顶点割集仅在图结构方面实现最小数目,没有考虑节点附加价值。

3 基于关键节点更新的关键路径分析

3.1 攻击图定义

攻击图是一种利用可获取资源进行多步攻击过程展示的建模方法。对于从初始节点开始,到达某一目的节点的攻击路径,可拆分为多步攻击行为。每一步攻击作为原子动作,完成攻击动作后可获得相应权限,以此条件进行下一步攻击动作,多步攻击行为级联达到最终攻击目标。攻击图通过定性分析,展示了目标网络中的多种威胁途径,结合量化计算可对威胁程度进行精确评估。

定义 1 攻击图为一个有向无环图,可表示

为 $AG = \{S, E, Pr\}$, 其中, S 表示攻击图中节点集合, $S = \{A, D\}$, A 表示原子攻击, D 表示设备路径二元组, 表示为 $D = \{src, dst\}$ 。其中, src 表示动作起始设备, dst 表示动作目标设备。 N 表示漏洞利用节点, 每进行一次攻击动作后, 可获取的设备权限。对于漏洞利用节点, 其包含 3 种类型, 即 $N = N_b \cup N_m \cup N_t$ 。其中, N_b 为初始节点集合, 定义为 $N_b = \{n_b \in N_b | \nexists n_j \in N, n_j \in Pa(n_b)\}$ 。 $Pa(n_b)$ 表示 n_b 的父节点集合, 定义某节点的父节点为指向该点的所有节点集合。 N_m 为中间节点集合, 定义为 $N_m = \{n_m \in N_m | \forall n_m \in N_m, Pa(n_m) \in N \wedge Ch(n_m) \in N\}$ 。其中, $Ch(n_m)$ 表示 n_m 的子节点集合, 定义其子节点为该点指向的所有节点集合。 N_t 为目标设备资源节点集合, 定义为 $N_t = \{n_t \in N_t | \nexists n_j \in N, n_j \in Ch(n_t)\}$ 。 E 表示连接节点的有向边集合, 其含义是只有达到攻击行为前置条件后, 才可触发某一攻击行为; 同时, 某一攻击行为发生后, 导致的设备权限的改变 (称为后置条件), 满足下一步原子攻击的条件状态后, 可执行下一步攻击。 Pr 为攻击行为发生的概率, $Pr \in [0, 1]$ 。

3.2 贝叶斯攻击图概率计算

本文贝叶斯攻击图中每个节点对应一个漏洞利用, 取值为 0 或 1。取值为 0 表示对应的漏洞未被利用, 取值为 1 表示对应的漏洞被利用。该节点被利用的概率为

$$\Pr(x_i = 1) = p \quad (1)$$

其中, x_i 是贝叶斯攻击图节点对应的漏洞, $\Pr(x_i = 1)$ 表示该节点被利用的概率, p 是概率值, 取值范围为 0 到 1。

定义在已知节点的父节点的状态下, 节点被利用的条件概率为 $p(x_i | Pa(x_i))$ 。如果该节点被利用的前提条件是其所有父节点都被利用, 则条件概率计算如式(2)所示。

$$p(x_i | Pa(x_i)) = \begin{cases} 0, \exists x_j \in Pa(x_i) \cap x_j = 0 \\ \prod_{j: x_j} p_{ij}, \text{其他} \end{cases} \quad (2)$$

其中, p_{ij} 是从父节点 x_j 利用子节点 x_i 的概率。

当存在一个父节点没有被利用, 则该节点就不可能被利用, 因此条件概率为 0。否则, 条件

概率为从所有父节点利用该节点的概率乘积。

如果该节点被利用的前提条件是至少有一个父节点被利用, 则条件概率的计算公式如下:

$$p(x_i | Pa(x_i)) = \begin{cases} 0, \forall x_j \in Pa(x_i) \cap x_j = 0 \\ 1 - \prod_{j: x_j} (1 - p_{ij}), \text{其他} \end{cases} \quad (3)$$

当该节点的所有父节点都没有被利用, 该节点才不可能被利用, 条件概率为 0。否则, 条件概率为父节点未能利用该节点的概率乘积的补。

贝叶斯攻击图中多个漏洞利用同时发生的概率为该攻击的漏洞利用组合的联合概率。被定义为在节点的父节点被攻击者利用后, 该节点被利用的条件概率的乘积:

$$P(X) = \prod_{i=1}^n p(x_i | Pa(x_i)) \quad (4)$$

其中, $X = \{x_1, x_2, \dots, x_n\}$ 对应贝叶斯攻击图中一组漏洞利用。

利用贝叶斯攻击图的条件概率以及全概率公式推导出节点的边概率, 即攻击者成功攻击节点的概率, 如式(5)所示。

$$p(x_i) = \sum_{X-x_i} \prod_{j=1}^n p(x_j | Pa(x_j)) \quad (5)$$

其中, X 表示攻击图中所有节点对应的漏洞利用的集合, $X - x_i$ 表示除去漏洞利用 x_i 的集合。

本文假设所有漏洞节点独立利用, 即只要存在成功利用的父节点, 子节点即满足利用条件。在实际情况下, 对于同一设备, 其利用某一漏洞取得权限后, 权限不会丢失。因此, 当某一漏洞成功利用后, 攻击者即可获得该设备相应的权限, 而不需要其余漏洞作为辅助; 当多个漏洞均可利用时, 选取该设备所有漏洞中攻击影响最大的漏洞, 即攻击者在该设备上获取的权限。

3.3 基于割集的攻击图关键节点选取算法

对于静态攻击图概率的计算, 无法考虑到网络环境的更新以及确定意向攻击对攻击图概率的影响。因此, 目前针对此问题提出的动态贝叶斯攻击图的解决办法, 但其存在计算效率较低的问题。

经研究分析发现, 工控环境中并非所有节点都属于关键节点。实际情况中, 无论是攻击者还

是防御者，都主要关注几个关键节点，对其进行攻击或重点防护。因此，本文利用动态更新思想结合关键节点集计算，改进静态攻击图的一次性计算问题。一般攻击图静态概率计算根据节点概率来单向计算节点关键性，本文考虑到全局网络对单一节点的影响，利用原始节点概率计算节点关键性，获得攻击图关键节点集合，并根据节点关键性反向更新原始节点概率。同时，针对确定意向的攻击行为，重新量化意向可达节点的节点概率。攻击图关键节点选取算法如算法1所示。

算法1 攻击图关键节点选取算法

输入 贝叶斯攻击图的点和边、初始节点、目标节点、漏洞利用节点概率
输出 攻击图关键节点集合

```

1) function UpdateCriticalNodesProbs (edges ,
bayNodes, startExp, goalExp, expProbs)
2)  paths, nodesCounter, nodesSorted, criticalNodes ← [], [], [], []
3)  freqDict ← {}
4)  paths, nodesCounter ← PathTraversal
(edges, bayNodes, startExp)
5)  delete startExp and goalExp from nodesCounter
6) foreach nodeCounter in nodesCounter do
7)    if nodeCounter.freq not in freqDict
then
8)      freqDict[nodeCounter.freq] ← []
9)      Add (nodeCounter, freqDict[nodeCounter.freq])
10) foreach key in Sort(freqDict.keys()) do
11)   if Length(freqDict[key]) > 0 then
12)     temp ← []
13)     foreach node in freqDict[key] do
14)       Income ← (node.bayProb *
node.cvss Score)
15)       node.freq ← income
16)       Add(node, temp)
17)     sortTemp ← Sort(temp)
18)     Add(sortTemp, nodesSorted)
19)   Continue
20)   Add(freqDict[key], nodesSorted)

```

```

21)     tempPaths ← paths
22) foreach nodes in nodesSorted do
23)   foreach node in nodes do
24)     if Length(tempPaths) = 0 do
25)       Break
26)     paths ← tempPaths
27)     foreach p in paths do
28)       if node in p do
29)         delete p from tempPaths
30)       if node not in criticalNodes do
31)         Add(node, criticalNodes)
32) UpdateNewProbs(criticalNodes, expProbs)
33) end function

```

算法1的2)~3)为初始化变量，首先遍历贝叶斯攻击图获得所有路径和每个节点在路径中出现的频率(4))，然后去除攻击图的开始和目标节点(5))。对所有节点进行统计，创建字典 freqDict，其键为频率，值为所有具有相同频率的节点(6)~9))。接下来10)~20)对具有相同频率的节点计算攻击收益(14))替换频率(15))，根据攻击收益进行排序(17)~18))，确定要选取的顺序。节点出现频率不同，则选取频率较大节点，最后变量 nodesSorted 的每个元素是根据原始的频率进行排序，每个元素又是一组频率相同的节点，其根据攻击收益进行排序。21)~31)是根据之前的排序从高到低选取节点，删除具有该节点的路径并将其添加到关键节点集(27)~31))，直到所有路径被删除(24)~25))最后更新关键节点的概率(32))。

利用频率相同的节点在路径覆盖度层面关键程度相同，需要进一步判断节点本身的重要性。本文参考最小顶点割集思想，对关键节点进行选取。攻击图的最小顶点割集是指将一个连通图变成两个连通分量所需删除的最少的点的集合。该思想仅在图的路径层面对割集进行选取，其假设所有点的权重价值均相等。

本文采取比较攻击收益方式进行割集节点选取。攻击收益的定义如下。

定义2 攻击收益指攻击者在利用漏洞完成对某一节点的攻击后，所能获得的收益。可表示为： $\text{Inc}(x_i) = \text{Pr}(x_i) \cdot \text{Sc}(x_i)$ 。其中， $\text{Inc}(x_i)$ 表示节点 x_i 的攻击收益， $\text{Pr}(x_i)$ 表示该节点的漏洞攻击

概率, $Sc(x_i)$ 表示漏洞危害得分。对于攻击收益的量化, 本文采用贝叶斯攻击图的原始漏洞攻击概率与通用漏洞评分系统 (CVSS) 漏洞危害最终评分进行计算。在漏洞危害相同的情况下, 漏洞攻击概率增大, 其攻击收益随之增加; 在漏洞攻击概率相同情况下, 漏洞危害越大, 其攻击收益越大。

3.4 基于灰色关联度分析法的关键节点概率计算

CVSS 漏洞评分的基础评分为普适性评分, 然而环境对漏洞危害性的影响不可忽略。尤其在安全程度越来越高的工控网络中, 其漏洞的利用环境以及影响程度不同于传统网络。本文对 3.3 节算法计算出的攻击图关键节点集合, 参考文献[14]中使用的量化方法, 对影响指标的选取以及客观性进行优化。利用灰色关联度分析法对关键漏洞节点的概率进行重新计算, 考虑工控关键影响因素之间的联系, 以及各种因素对漏洞利用节点的影响, 精确量化关键漏洞节点在工控环境下的概率变化。

3.4.1 计算指标

文献[14]中防御强度、攻击强度等量化指标需要参考大量专家经验作为计算依据, 本文采用 CVSS 中各项漏洞指标对关键漏洞节点的概率进行更新计算, 漏洞攻击概率定义如下。

定义 3 漏洞攻击概率通常受漏洞可利用性以及漏洞影响程度两个指标影响 (详细指标度量如表 1 所示), 可表示为 $Pr_{new}(x_i) = Ex(x_i) \cdot Im(x_i)$ 。其中, $Pr_{new}(x_i)$ 表示节点 x_i 的更新攻击概率, $Ex(x_i)$ 表示该节点的漏洞可利用性, $Im(x_i)$ 表示漏洞影响程度 (Impact)。

(1) 漏洞可利用性

漏洞可利用性指某一漏洞被攻击者利用从而

进行攻击的可能性, 是漏洞的固有属性。当漏洞可利用性增大时, 该漏洞的攻击概率也会增大, 漏洞可利用性与漏洞攻击概率呈正相关。

漏洞可利用性根据 CVSS 中的基本指标进行量化计算。CVSS 中对漏洞可利用性的评分分为 3 个方面: 访问向量 (AV)、访问复杂度 (AC)、身份认证 (AU) [22]。

(2) 漏洞影响程度

漏洞影响程度指当某一漏洞被成功利用后, 其对漏洞所在环境造成的破坏或影响。当漏洞影响程度增大时, 该漏洞的攻击概率同样也会增大, 漏洞影响程度与漏洞攻击概率成正相关。

漏洞影响程度由系统性能影响与工控设备资产价值两方面进行量化。系统性能影响分为 3 个方面: 机密性影响 (C)、完整性影响 (I)、可用性影响 (E) (如表 1 所示) [22]。资产价值 (DV) 则根据设备所在的不同层次, 以及每种设备的执行功能、控制范围、与其余设备关联程度的不同情况所决定。设备对系统影响程度越大, 该设备价值越高。本文列出了常见工控设备的影响程度, 如表 2 所示。

表 2 工控设备的影响程度
Table 2 Industrial control equipment impact

影响程度	设备类型
1	文件服务器、HMI
2	OPC 服务器
3	SCADA、操作员站、PLC
4	数据库、工程师站

3.4.2 灰色关联度分析法

两个因素随时间或不同对象而变化的关联性大小的量度, 被称为关联度。灰色关联度分析法

表 1 漏洞概率影响因素
Table 1 Vulnerability probability influencing factors

影响因素	指标	等级划分	分值	权重
可利用性	访问向量	本地 (L) / 相邻网络 (A) / 网络 (N)	0.395/0.646/1.0	0.34
	访问复杂度	高 (H) / 中等 (M) / 低 (L)	0.35/0.61/0.71	0.33
	身份认证	多个 (M) / 单个 (S) / 无 (N)	0.45/0.56/0.704	0.33
	资产价值	弱小 (L) / 一般 (M) / 严重 (H) / 重大 (S)	0.2/0.55/0.62/0.85	0.4
影响程度	机密性影响	无 (N) / 部分 (P) / 完成 (C)	0.0/0.275/0.66	0.2
	完整性影响	无 (N) / 部分 (P) / 完成 (C)	0.0/0.275/0.66	0.2
	可用性影响	无 (N) / 部分 (P) / 完成 (C)	0.0/0.275/0.66	0.2

作为衡量因素间关联程度的一种方法，可对上述多种影响工控环境的因素进行关联度量化，以此作为考虑多方面因素对关键漏洞节点进行概率更新计算的依据。灰色关联度分析法计算过程如下。

首先，设有 n 个被评价目标，每个被评价目标有 p 个评价指标。则第 i 个对象描述为

$$x_i = (x_{i1}, x_{i2}, \dots, x_{ip}) \quad (6)$$

在各项指标中选出最优值组成参考序列 x_0 ：

$$x_0 = (x_{01}, x_{02}, \dots, x_{0p}) \quad (7)$$

关联系数 ζ_i 表示第 i 个评价目标与参考序列间的关系：

$$\zeta_i(k) = \frac{\min_i \min_k |x_0(k) - x_i(k)| + \rho \cdot \max_i \max_k |x_0(k) - x_i(k)|}{|x_0(k) - x_i(k)| + \rho \max_i \max_k |x_0(k) - x_i(k)|} \quad (8)$$

其中， ρ 为分辨系数。

然后计算关联度 γ_{0i} ， γ_{0i} 表示各指标与参考序列间的关联关系，计算如下：

$$\gamma_{0i} = \frac{1}{p} \sum_{k=1}^p W_k \delta_{ij}, i=1, 2, \dots, n \quad (9)$$

其中， W_k 为权重， $W_k \in (0, 1)$ 。

最后，利用定义 3 计算出更新后节点的漏洞利用概率。

3.5 针对部分更新攻击图的关键路径选取算法

动态贝叶斯攻击图考虑全局节点环境的实时变化情况，并对其进行概率更新。由于工控网络节点较多，且节点间关联关系较为复杂，全局节点动态更新效率较低。

本文针对 3.3 节算法获得的关键节点集合，利用 3.4 节的概率计算方法，只更新关键节点的漏洞利用概率。遍历攻击图中全部攻击路径，重新计算得出每条攻击路径概率，比较得出最大攻击概率的路径，并将其作为攻击图的关键路径。

部分更新的贝叶斯攻击图关键路径概率计算如算法 2 所示。

算法 2 部分更新的贝叶斯攻击图关键路径概率计算算法

输入 所有漏洞利用节点、初始节点、目标节点、漏洞利用节点概率

输出 贝叶斯攻击图、所有路径概率

```

1) function GenerateAttackGraph (exploits,
start-Exp, goalExp, expProbs)
2) queueExps  $\leftarrow$  Queue( $\emptyset$ )
3) edges, nodes, edges_re, nodes_re, visited,
pathsProb  $\leftarrow$  [], [], [], [], [], []
4) Enqueue(startExp, queueExps)
5) while queueExps is not empty do
6)   currExp  $\leftarrow$  Dequeue(queueExps)
7)   Add(currExp, nodes)
8)   currPriv  $\leftarrow$  currExp.postcond
9)   foreach exploit in exploits do
10)    precond  $\leftarrow$  exploit.precond
11)    if Matching(currPriv, precond) then
12)      Create_edge(currExp, exploit, edges)
13)      if exploit is not in visited then
14)        Enqueue(exploit, queueExps)
15)        Add(exploit, visited)
16) BreakCircles(edges, nodes)
17) visited  $\leftarrow$  []
18) Enqueue(goalExp, queueExps)
19) while queueExps is not empty do
20)   currExp  $\leftarrow$  Dequeue(queueExps)
21)   Add(currExp, nodes_re)
22)   currPriv  $\leftarrow$  currExp.precond
23)   foreach exploit in nodes do
24)     postcond  $\leftarrow$  exploit.postcond
25)     if Matching(currPriv, postcond) then
26)       if edges include an edge from exploit to currExp then
27)         Create_edge (exploit, currExp,
edges_re)
28)         if exploit is not in visited then
29)           Enqueue(exploit, queue Exps)
30)           Add(exploit, visited)
31) bayNodes  $\leftarrow$  []
32) bayNodes  $\leftarrow$  CalBayesianProbs(edges,
nodes, expProbs)
33) expProbs  $\leftarrow$  UpdateCriticalNodesProbs
(edges, bayNodes, startExp, goalExp, expProbs)
34) bayNodes  $\leftarrow$  CalBayesianProbs(edges,

```

nodes, expProbs)

35) pathsProb $\leftarrow$ CalPathsProb(edges, bay-Nodes)

36) end function

算法的 2)~3)为初始化变量, 4)将开始节点加入队列, 5)~15)为正向广度优先遍历生成漏洞利用的向前依赖图, 6)出队列, 遍历从开始节点的所有可达节点, 对于当前节点, 遍历所有可能的漏洞利用 (9)), 然后通过判断当前节点的后置条件 (8)) 是否能满足遍历的漏洞利用的前置条件 (11)), 如果满足则加边 (12)), 如果该漏洞没有利用过, 则将其加入队列, 并标记访问 (13)~15)。获得向前依赖图后, 需要进行去环, 去除重复访问的边 (16))。然后再从目标节点开始逆向广度优先遍历之前生成的向前依赖图 (行 17)~30)) 生成攻击图, 这样能保证图中所有节点从开始和目标节点都可达。获得原始攻击图后, 根据漏洞利用概率计算贝叶斯攻击图

(32)), 然后利用关键节点算法更新漏洞利用的概率 (33)) 后, 再计算更新概率后的贝叶斯攻击图 (34)), 最后遍历贝叶斯攻击图, 根据节点贝叶斯概率获得攻击图路径概率 (35))。

4 实验和分析

4.1 实验拓扑环境

本文采用文献[23]的拓扑结构来举例说明本文所提方法的工作原理, 并对该方法进行验证。实验采用的拓扑结构源于实际工控网络中的架构, 增加了结构的合理性以及真实性, 但对规模进行了限制。此外, 该拓扑结构增加了不同类型的工控设备, 并为不同设备设置了常见的漏洞, 以使实验结果更加全面, 设备漏洞情况如表 3 所示。

实验拓扑图如图 1 所示。实验室采用的拓扑结构由 4 个区域组成: 外部区域、DMZ 区、监视控制区、流程操作区。外部区域设置了一个远程

表 3 设备漏洞情况
Table 3 Device vulnerability

设备名称	漏洞编号	访问	脆弱软件	影响因素分值	CVSS2.0 漏洞评分
ew2	CVE-2014-5424	远程访问	CCW	{AV:N/AC:L/AU:N}{/DV:H/C:P/I:P/A:P}	7.5
ew1	CVE-2014-8551a	远程访问	TIA Portal	{AV:N/AC:L/AU:N}{/DV:H/C:C/I:C/A:C}	10.0
scada	CVE-2014-8551b	远程访问	WinCC	{AV:N/AC:L/AU:N}{/DV:H/C:C/I:C/A:C}	10.0
historian	CVE-2011-4034	受害设备	Vijeo Historian	{AV:N/AC:M/AU:N}{/DV:S/C:C/I:C/A:C}	9.3
PLC1	CVE-2015-2177	远程访问	SIMATIC S7-300 CPU	{AV:N/AC:L/AU:N}{/DV:H/C:N/I:N/A:C}	7.8
PLC2	CVE-2015-6490	远程访问	Micrologix 1400	{AV:N/AC:L/AU:N}{/DV:H/C:C/I:C/A:C}	10.0
PLC3	CVE-2014-0754	远程访问	SchneiderWeb	{AV:N/AC:L/AU:N}{/DV:H/C:C/I:C/A:C}	10.0
ew1	CVE-2014-6332	受害设备	MS Windows	{AV:N/AC:M/AU:N}{/DV:H/C:C/I:C/A:C}	9.3

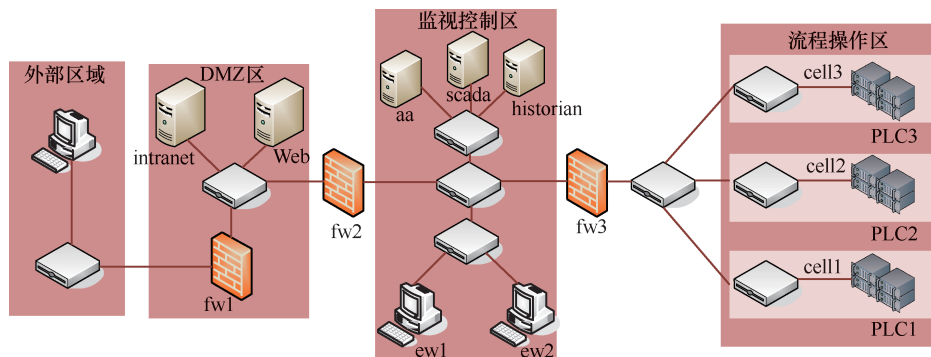


图 1 实验拓扑图

Figure 1 Experimental topology graph

节点 i , 通过该节点攻击者可以登录并开始指定目标的攻击。DMZ 区与外部区域的连接由防火墙 fw1 进行保护。DMZ 区中包含两个服务器, 其允许访问公司内部网络。监视控制区由防火墙 fw2 连接到 DMZ 区, 包括服务器和 workstation, 用于控制流程操作区的现场设备, 完成数据收集统计及传输^[23]。其中, ew1 和 ew2 作为两个 workstation, 分别配置了西门子工作台与罗克韦尔工作台。流程操作区由 3 个执行单元 (cell) 构成, 包括可编程逻辑控制器 (PLC) 以及现场设备。本拓扑的 3 个执行单元分别来自不同供应商: 西门子、罗克韦尔、施耐德。具体设备访问关系如表 4 所示。

表 4 设备访问关系
Table 4 Device access relationship

设备名称	可访问设备列表
i	Web(Webvul), intranet(MS Windows)
intranet	ew1(TIA Portal, MS Windows), ew2(CCW), Web(Webvul)
Web	ew1(TIA Portal, MS Windows), ew2(CCW)
ew1	scada(WinCC), PLC1(SIMATIC S7-300)
ew2	ew1(TIA Portal), scada(WinCC), PLC2(Micrologix 1400)
scada	historian(Vijeo Historian), PLC1(SIMATIC S7-300), PLC2(Micrologix 1400), PLC3(SchneiderWEB)
historian	PLC1(SIMATIC S7-300), PLC2(Micrologix 1400), PLC3(SchneiderWEB)
PLC1	scada(WinCC)
PLC2	scada(WinCC)
PLC3	scada(WinCC)

4.2 实验结果展示及分析

本文依照实验拓扑结构以及设备漏洞关系, 利用广度优先搜索算法, 以 PLC 为攻击者目标节点, 生成带有贝叶斯原始概率的攻击图, 如图 2 所示。节点 “MS Windows(Web, ew1) (0.697 5)” 的含义是攻击者在获取到 Web 权限后, 利用 ew1 上 MS Windows 所包含的漏洞, 获取 ew1 上的权限。0.697 5 表示该漏洞利用的发生概率, 即从 Web 利用 MS Windows 成功到达 ew1 的概率。

4.2.1 关键节点选取算法有效性分析

两种算法下的关键顶点集如图 3 所示。其中, 图 3(a) 是利用文献[21]中的算法计算出的割集, 包含 3 个漏洞利用节点: {CCW(Web, ew2) (0.66), WinCC(ew1, scada) (0.673 9), WinCC(PLC1, scada) (0.667 9)}; 图 3(b) 是利用本文提出的算法计算出的割集, 包含 4 个漏洞利用节点: {CCW(Web, ew2) (0.66), WinCC(ew1, scada) (0.673 9), WinCC(PLC1, scada) (0.667 9), Schneider Web(scada, PLC3) (0.735 4)}。利用本文提出的算法计算出的关键节点集包含了利用文献[21]算法计算出的集合中的所有节点, 并且多了一个漏洞利用节点 Schneider Web(scada, PLC3) (0.735 4)。在实际环境中分析, 该漏洞利用节点的目标权限获取设备为 PLC3, PLC 为工控网络中较为重要的设备, 一旦 PLC 被攻击者攻击, 会直接对控制逻辑以及底层现场设备产生直接影响。同时, 该漏洞利用的概率为 0.735 4, 相比于一般节点, 其有较高的被利用概率。本次利用的漏洞为 PLC3 上

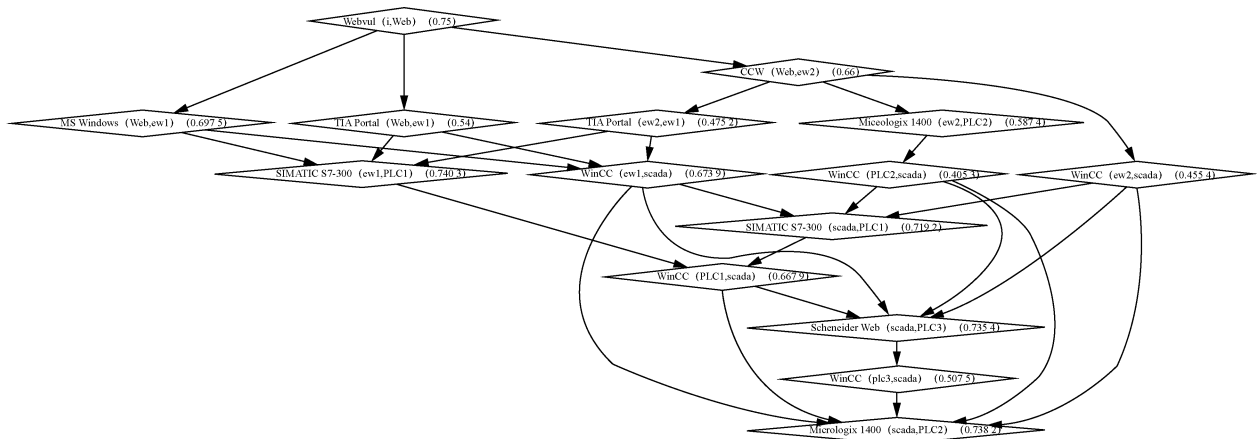


图 2 贝叶斯原始攻击概率攻击图
Figure 2 Bayes primitive probability attack graph

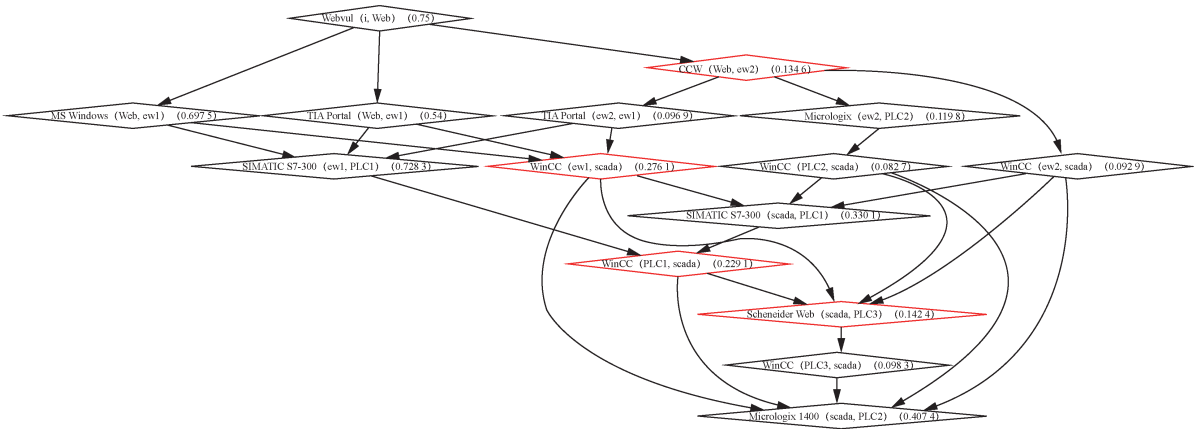
Schneider Web 包含的漏洞 CVE-2014-0754, 其 CVSS2.0 评分 (由于 CVSS2.0 覆盖较多漏洞的详细评分, 因此本文选用 CVSS2.0 标准为参考量化指标) 为满分 10 分, 意味着该漏洞极其危险。一旦远程攻击者成功利用漏洞, 其可通过发送特制的 HTTP 请求利用访问任意资源。因此, 从攻击概率、攻击影响方面考虑, 该点的攻击收益很高,

将该点设置为关键节点。而文献[21]以及文献[24]中算法计算出的 3 个节点, 只从图结构方面考虑, 仅计算最小顶点割集, 但在节点价值各不相同的工控环境下, 其可能漏掉非最小割集中的关键节点。

表 5 列出了攻击图节点数从 21~249 的 5 种规模的攻击图关键节点计算的算法性能。本文选取



(a) 最小割集算法关键节点集合



(b) 本文关键节点集合

图 3 两种算法下的攻击图关键节点集合

Figure 3 Key node set of the attack graph under the two algorithms

表 5 关键节点算法性能
Table 5 Key node algorithm performance

规模	拓扑节点	拓扑边	攻击图节点	攻击图边	本文算法		顶点割集算法 ^[21]		矩阵割集算法 ^[24]	
					计算耗时	占用内存	计算耗时	占用内存	计算耗时	占用内存
规模 1	13	35	21	40	0.002 448 s	22.1 MB	0.002 472 s	21.9 MB	0.000 855s	21.9 MB
规模 2	31	101	59	120	0.008 824 s	22.2 MB	0.008 523 s	22.2 MB	1.547 3 s	22.1 MB
规模 3	49	167	97	200	0.017 48 s	22.4 MB	0.034 8 s	22.4 MB	8 167.159 2 s	22.5 MB
规模 4	85	299	173	360	0.040 39 s	22.7 MB	475.383 7 s	26.5 MB	—	—
规模 5	121	431	249	520	0.071 51 s	23.0 MB	52 h	13.449G	—	—

算法耗时、占用内存两种性能指标进行对比实验。从表中数据可知，在 173 个节点规模之前，本文算法耗时和占用内存略小于文献[21]提出的算法（顶点割集）。在 249 个节点前，文献[24]算法（矩阵割集）消耗内存较小，小规模计算的内存消耗优于本文算法。但在 249 个节点的规模时，顶点割集算法占用内存从 173 个节点时的 26.5 MB 爆炸增长至 13.45 GB。而本文算法的内存占用量增长幅度仍较小，整体呈线性增长。

在算法耗时方面，顶点割集算法在 21、59 个节点规模时，其耗时表现与本文的算法基本持平。但是在 97 个节点时，其计算时间是本文算法的两倍。从 97 个节点开始的 3 个规模攻击图中，顶点割集算法计算耗时呈指数增长，在 173 个节点时，耗时高达 475 s，而本文算法耗时 0.04 s。在 249 个节点时，其耗时由计算估计约 52 h，本文算法计算耗时仍小于 0.1 s。矩阵割集算法在 21 个节点规模下表现出优异的计算耗时，优于本文算法以及顶点割集算法，但随着节点数的增长，该方法耗时呈爆炸式增长，97 个节点时耗时超过 2 h，173 个节点耗时无法经过实验计算，不适用于大规模计算。

4.2.2 关键路径选取策略有效性分析

根据 4.2.1 节实验得到的 4 个关键节点，利用 3.4 节所提出的算法进行概率计算，并更新节点概率。更新前节点概率如图 2 所示，更新后的节点概率

率如图 4 所示。在更新完关键节点集合中所有节点概率后，利用式(5)重新计算出每条攻击路径的攻击概率。列出了以 PLC2 为直接攻击节点的攻击路径如表 6 所示。

表 6 攻击路径概率
Table 6 Attack path probability

攻击路径	更新前概率	更新后概率
$i \rightarrow \text{Web} \rightarrow \text{ew1(M)} \rightarrow \text{scada} \rightarrow \text{PLC2}$	0.260	5.89×10^{-2}
$i \rightarrow \text{Web} \rightarrow \text{ew1(T)} \rightarrow \text{scada} \rightarrow \text{PLC2}$	0.201	4.56×10^{-2}
$i \rightarrow \text{Web} \rightarrow \text{ew2} \rightarrow \text{scada} \rightarrow \text{PLC2}$	0.166	3.82×10^{-3}
$i \rightarrow \text{Web} \rightarrow \text{ew2} \rightarrow \text{ew1(T)} \rightarrow \text{scada} \rightarrow \text{PLC2}$	0.117	1.10×10^{-3}

如表 7 所示，本文采用对全局节点更新概率计算关键路径与只更新关键节点概率计算关键攻击路径两种更新策略进行计算耗时以及占用内存两方面对比实验。在 21 个攻击图节点到 1 161 个攻击图节点中设置了 9 种规模的攻击图实验。在占用内存方面，两种策略表现基本相同，且对于节点数大幅增长的情况，占用内存数没有较大幅增长。

在计算耗时方面，本文用折线图的方式直观地展示两种策略的耗时情况，如图 5 所示。从图 5 可以看出两种策略随着节点数不断增大，其耗时呈线性增长。但全局节点更新时间增长函数的斜率明显大于部分节点更新。由于两种策略的计算耗时相差数值在攻击图节点数较小时并不明

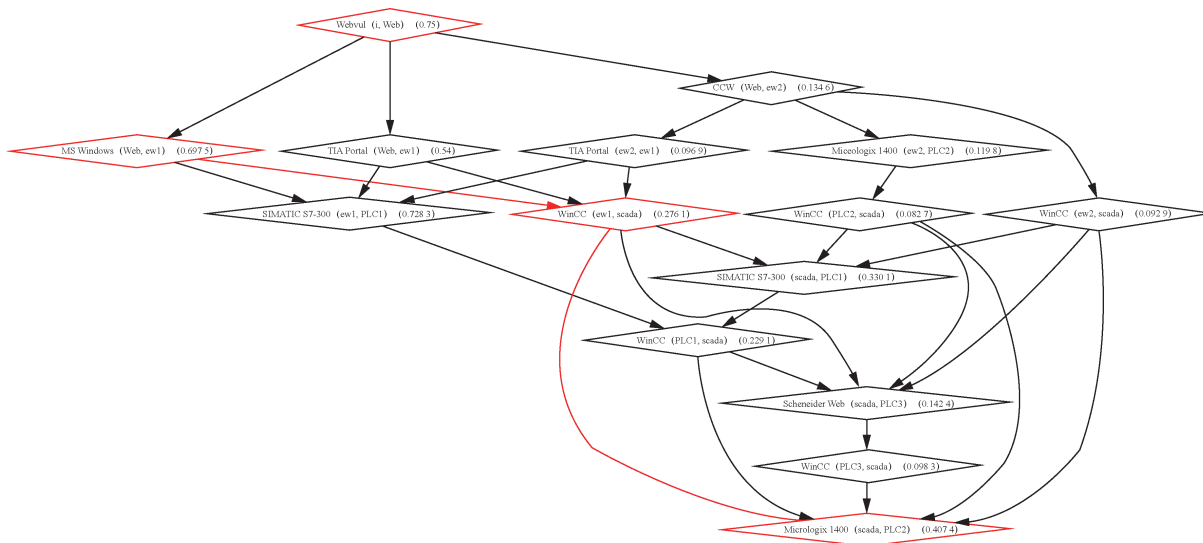


图 4 最大攻击路径概率攻击图
Figure 4 Maximum attack path probability attack graph

表7 更新节点策略性能
Table 7 Update node strategy performance

规模	攻击图节点	全局更新时间/s	占用内存/MB	部分更新时间/s	占用内存/MB
规模 1	21	0.026 88	22.0	0.002 12	22.0
规模 2	59	0.077 26	22.1	0.006 28	22.1
规模 3	97	0.130 4	22.3	0.010 48	22.3
规模 4	173	0.233 35	22.7	0.018 21	22.6
规模 5	249	0.332 5	22.9	0.026 15	23.0
规模 6	401	0.546 8	23.6	0.042 05	23.5
规模 7	553	0.754 8	24.2	0.057 21	24.0
规模 8	857	1.157 5	25.3	0.087 92	25.3
规模 9	1161	1.573 4	26.8	0.115 1	26.8

显, 在 21 个节点时, 全局节点更新耗时 0.026 88 s, 部分节点更新耗时为 0.002 119 s, 相差 0.024 761 s, 但全局更新耗时约为部分节点更新耗时的 13 倍。在 1 161 个节点时, 全局节点更新耗时 1.573 4 s, 部分节点更新耗时为 0.115 1 s, 全局更新耗时仍约为部分节点更新耗时的 14 倍, 但其耗时相差 1.458 3 s。经数据分析可知, 本文的部分更新策略在攻击图规模越大时, 效果越好。本文在关键路径计算结果与传统算法相同的情况下, 耗时更低, 可应用于大规模工控网络的关键路径计算中。

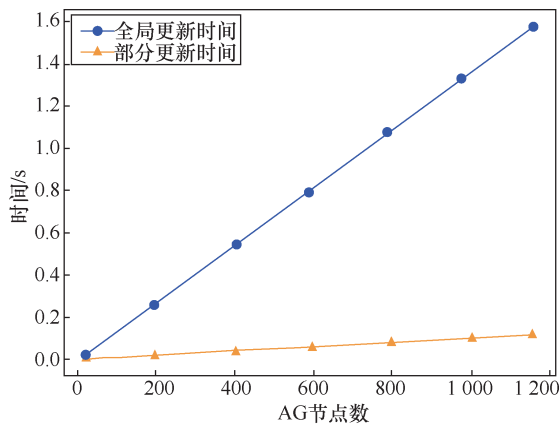


图5 全局/部分更新算法耗时对比

Figure 5 Full/partial update algorithm time-consuming comparison

5 结束语

本文主要研究大规模工控网络下攻击图的关键路径选取问题, 并提出了一种基于攻击收益割集选取关键节点集合的算法, 同时提出了以部分更新关键节点攻击概率代替全局更新节点概率的策略。本文结合实际案例对计算过程进行分析, 同时通过

实验对比, 验证了所提方法在大规模工控攻击图中的计算效率相比于最小割集算法以及全局节点更新策略有大幅提升, 同时保证了方法结果的一致性。

未来的研究工作包括: 对于全局攻击图的小概率攻击路径简化; 对于无实际意义的多条重复路径的优化。

参考文献:

- [1] NOURIAN A, MADNICK S. A systems theoretic approach to the security threats in cyber physical systems applied to Stuxnet[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 15(1): 2-13.
- [2] LIANG G Q, WELLER S R, ZHAO J H, et al. The 2015 Ukraine blackout: implications for false data injection attacks[J]. IEEE Transactions on Power Systems, 2017, 32(4): 3317-3318.
- [3] AKBANOV M, VASSILAKIS V G, LOGOTHETIS M D. Ransomware detection and mitigation using software-defined networking: the case of WannaCry[J]. Computers & Electrical Engineering, 2019, 76: 111-121.
- [4] ALLADI T, CHAMOLA V, ZEADALLY S. Industrial control systems: cyberattack trends and countermeasures[J]. Computer Communications, 2020, 155: 1-8.
- [5] 吴晨思, 谢卫强, 姬逸潇, 等. 网络系统安全度量综述[J]. 通信学报, 2019, 40(6): 14-31.
WU C S, XIE W Q, JI Y X, et al. Survey on network system security metrics[J]. Journal on Communications, 2019, 40(6): 14-31.
- [6] PHILLIPS C, SWILER L P. A graph-based system for network-vulnerability analysis[C]//Proceedings of the 1998 Workshop on New Security Paradigms - NSPW '98. 1998: 71-79.
- [7] 叶子维, 郭渊博, 王宸东, 等. 攻击图技术应用研究综述[J]. 通信学报, 2017, 38(11): 121-132.
YE Z W, GUO Y B, WANG C D, et al. Survey on application of attack graph technology[J]. Journal on Communications, 2017, 38(11): 121-132.
- [8] POOLSAPPASIT N, DEWRI R, RAY I. Dynamic security risk management using Bayesian attack graphs[J]. IEEE Transactions

- on Dependable and Secure Computing, 2012, 9(1): 61-74.
- [9] MUÑOZ-GONZÁLEZ L, SGANDURRA D, BARRÈRE M, et al. Exact inference techniques for the analysis of Bayesian attack graphs[J]. IEEE Transactions on Dependable and Secure Computing, 2017, 16(2): 231-244.
- [10] 杨英杰, 冷强, 常德显, 等. 基于属性攻击图的网络动态威胁分析技术研究[J]. 电子与信息学报, 2019, 41(8): 1838-1846.
YANG Y J, LENG Q, CHANG D X, et al. Research on network dynamic threat analysis technology based on attribute attack graph[J]. Journal of Electronics & Information Technology, 2019, 41(8): 1838-1846.
- [11] 王辉, 王云峰, 王坤福. 基于贝叶斯推理的攻击路径预测研究[J]. 计算机应用研究, 2015, 32(1): 226-231.
WANG H, WANG Y F, WANG K F. Research on predicting attack path based on Bayesian inference[J]. Application Research of Computers, 2015, 32(1): 226-231.
- [12] 张少俊, 李建华, 宋珊珊, 等. 贝叶斯推理在攻击图节点置信度计算中的应用[J]. 软件学报, 2010, 21(9): 2376-2386.
ZHANG S J, LI J H, SONG S S, et al. Using Bayesian inference for computing attack graph node beliefs[J]. Journal of Software, 2010, 21(9): 2376-2386.
- [13] 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 model[J]. Chinese Journal of Computers, 2014, 37(1): 62-72.
- [14] 黄家辉, 冯冬芹, 王虹鉴. 基于攻击图的工控系统脆弱性量化方法[J]. 自动化学报, 2016, 42(5): 792-798.
HUANG J H, FENG D Q, WANG H J. A method for quantifying vulnerability of industrial control system based on attack graph[J]. Acta Automatica Sinica, 2016, 42(5): 792-798.
- [15] 罗智勇, 杨旭, 刘嘉辉, 等. 基于贝叶斯攻击图的网络入侵意图分析模型[J]. 通信学报, 2020, 41(9): 160-169.
LUO Z Y, YANG X, LIU J H, et al. Network intrusion intention analysis model based on Bayesian attack graph[J]. Journal on Communications, 2020, 41(9): 160-169.
- [16] 马春光, 汪诚弘, 张东红, 等. 一种基于攻击意愿分析的网络风险动态评估模型[J]. 计算机研究与发展, 2015, 52(9): 2056-2068.
MA C G, WANG C H, ZHANG D H, et al. A dynamic network risk assessment model based on attacker's inclination[J]. Journal of Computer Research and Development, 2015, 52(9): 2056-2068.
- [17] 高梦州, 冯冬芹, 凌从礼, 等. 基于攻击图的工业控制系统脆弱性分析[J]. 浙江大学学报(工学版), 2014, 48(12): 2123-2131.
GAO M Z, FENG D Q, LING C L, et al. Vulnerability analysis of industrial control system based on attack graph[J]. Journal of Zhejiang University (Engineering Science), 2014, 48(12): 2123-2131.
- [18] 叶云, 徐锡山, 齐治昌. 大规模网络中攻击图的节点概率计算方法[J]. 计算机应用与软件, 2011, 28(11): 136-139, 192.
YE Y, XU X S, QI Z C. Attack graph's nodes probabilistic computing approach in a large-scale network[J]. Computer Applications and Software, 2011, 28(11): 136-139, 192.
- [19] 刘贞宇, 陈羽中, 郭昆, 等. 面向网络攻击建模的分布式过程挖掘与图分割方法[J]. 小型微型计算机系统, 2020, 41(8): 1732-1740.
LIU Z Y, CHEN Y Z, GUO K, et al. Distributed process mining and graph segmentation for network attack modeling[J]. Journal of Chinese Computer Systems, 2020, 41(8): 1732-1740.
- [20] 吴金字, 金舒原, 杨智. 基于网络流的攻击图分析方法[J]. 计算机研究与发展, 2011, 48(8): 1497-1505.
WU J Y, JIN S Y, YANG Z. Analysis of attack graphs based on network flow method[J]. Journal of Computer Research and Development, 2011, 48(8): 1497-1505.
- [21] 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.
PENG W, HU C Z, YAO S P. Intrusive intention recognition based on attack path graph[J]. Transactions of Beijing Institute of Technology, 2010, 30(9): 1077-1081.
- [22] Common vulnerability scoring system version 2.0[EB].
- [23] CHEMINOD M, DURANTE L, SENO L, et al. Detection of attacks based on known vulnerabilities in industrial networked systems[J]. Journal of Information Security and Applications, 2017, 34: 153-165.
- [24] CHE Y B, ZHAO Y C, XU J M, et al. A hierarchical approach for fast calculating minimal cut sets of a microgrid[J]. Mathematical Problems in Engineering, 2017, 2017: 1-8.

[作者简介]



张耀方(1998-), 女, 黑龙江哈尔滨人, 哈尔滨工业大学(威海)博士生, 主要研究方向为工业控制系统安全风险

评估。



张哲宇(1989-), 女, 吉林松原人, 硕士, 国家工业信息安全发展研究中心工程师, 主要研究方向为网络安全、工业信息安全、关键信息基础设施保护、检测评估技术。

曲海阔(1997-), 女, 辽宁鞍山人, 哈尔滨工业大学(威海)硕士生, 主要研究方向为工业控制系统安全检测。

张格(1980-), 男, 北京人, 国家工业信息安全发展研究中心副主任, 主要研究方向为工业信息安全、网络安全检查评估技术、网络空间安全态势、关键信息基础设施网络安全防护技术。

王子博(1992-), 男, 黑龙江哈尔滨人, 哈尔滨工业大学(威海)博士生, 主要研究方向为工业控制系统安全风险

评估。

王佰玲(1978-), 男, 黑龙江哈尔滨人, 哈尔滨工业大学(威海)教授、博士生导师, 主要研究方向为网络安全、信息内容安全、信息对抗、工业互联网安全。