

基于网络防御知识图谱的 0day 攻击路径预测方法

孙澄, 胡浩, 杨英杰, 张红旗

(信息工程大学, 河南 郑州 450001)

摘要: 针对 0day 漏洞未知性造成的攻击检测难问题, 提出了一种基于知识图谱的 0day 攻击路径预测方法。通过从现有关于网络安全领域本体的研究成果及网络安全数据库中抽取“攻击”相关的概念及实体, 构建网络防御知识图谱, 将威胁、脆弱性、资产等离散的安全数据提炼为互相关联的安全知识。在此基础上, 依托知识图谱整合的知识, 假设并约束 0day 漏洞的存在性、可用性及危害性等未知属性, 并将“攻击”这一概念建模为知识图谱中攻击者实体与设备实体间存在的一种关系, 从而将攻击预测问题转化为知识图谱的链接预测问题。采用基于路径排序算法的知识图谱推理方法挖掘目标系统中可能发生的 0day 攻击, 并生成 0day 攻击图。复用分类器输出的预测得分作为单步攻击发生概率, 通过计算并比较不同攻击路径的发生概率, 预测分析 0day 攻击路径。实验证明, 所提方法能够依托知识图谱提供的知识体系, 为攻击预测提供较全面的知识支持, 降低预测分析对专家模型的依赖, 并较好地克服 0day 漏洞未知性对预测分析造成的不利影响, 提高了 0day 攻击预测的准确性, 并且借助路径排序算法基于图结构这一显式特征进行推理的特点, 能够对推理结果形成的原因进行有效反溯, 从而一定程度上提高了攻击预测分析结果的可解释性。

关键词: 知识图谱; 0day 攻击; 攻击路径预测

中图分类号: TP393.08

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2021101

Prediction method of 0day attack path based on cyber defense knowledge graph

SUN Cheng, HU Hao, YANG Yingjie, ZHANG Hongqi

Information Engineering University, Zhengzhou 450001, China

Abstract: To solve the difficulty of attack detection caused by the 0day vulnerability, a prediction method of 0day attack path based on cyber defense knowledge graph was proposed. The cyber defense knowledge graph was constructed to refine the discrete security data such as threat, vulnerability and asset into the complete and high-related knowledge format by extracting concepts and entities related to network attack from cyber security ontology research finds and databases. Based on the knowledge integrated by the knowledge graph, assumed and

收稿日期: 2021-01-07; 修回日期: 2021-03-06

通信作者: 孙澄, suncheng1991@outlook.com

基金项目: 国家自然科学基金 (61902427)

Foundation Item: The National Natural Science Foundation of China (61902427)

引用格式: 孙澄, 胡浩, 杨英杰, 等. 基于网络防御知识图谱的 0day 攻击路径预测方法[J]. 网络与信息安全学报, 2022, 8(1): 151-166.

Citation Format: SUNC, HU H, YANG Y J, et al. Prediction method of 0day attack path based on cyber defense knowledge graph[J]. Chinese Journal of Network and Information Security, 2022, 8(1): 151-166.

restricted the unknown attributes such as the existence, availability and harmfulness of 0day vulnerabilities, and model the concept of "attack" as a relationship between attacker entities and device entities in the knowledge graph to transform the attack prediction to the link prediction of knowledge graph. According to this, apply path ranking algorithm was applied to mine the potential 0day attack in the target system and construct the 0day attack graph. Predicted the 0day attack path by utilizing the scores output by classifiers as the occurrence probabilities of single step attack and computing the occurrence probabilities of different attack paths. The experimental result shows that with the help of complete knowledge system provided by knowledge graph, the proposed method can reduce the dependence of prediction analysis on expert model and overcome the bad influence of 0day vulnerability to improve the accuracy of 0day attack prediction. And utilizing the characteristic that path ranking algorithm reasons based on the structure of graph can also help to backtrack the reasons of predicting results so as to improve the explainability of predicting.

Keywords: knowledge graph, 0day attack, attack path prediction

0 引言

在当今网络攻防体系中, 0day 漏洞是指未被安全厂商发现, 却可能被黑客组织掌握的系统脆弱性的总称。在被用于发起 0day 攻击时, 因其对于防御者具有未知性, 造成攻防双方严重的信息不对称, 使防御者缺乏漏洞细节信息, 难以实施有效检测, 攻击者因此提高了攻击的隐蔽性及成功率。当前, 0day 攻击常被应用于入侵高防护级别的信息系统并实现攻击的隐蔽持续, 如“震网”事件。由此可见, 如何克服 0day 漏洞未知性导致的 0day 攻击难以检测的问题已成为当前网络安全领域面临的一大难题。

关于 0day 攻击检测的研究主要包括事前的攻击预测及融合告警的实时检测两方面, 其中, 攻击预测是指在攻击发生前, 综合系统配置信息及有关攻击知识, 对目标系统中潜在 0day 攻击的位置、可能性及影响进行预判。具体可分为基于统计学的预测技术及基于攻击场景的预测技术。前者通过统计分析现有漏洞的生命周期, 对新漏洞出现的时间、数量及宿主等进行预测, 进而实现 0day 攻击的预测^[1-2], 然而, 此类预测技术未结合攻击场景, 分析 0day 攻击对系统安全造成的危害, 且准确度不高。后者则以已知漏洞利用模型为参照, 构建 0day 攻击模型, 在生成已知攻击路径的基础上, 依据同一路径所属攻击的前后依赖关系, 匹配攻击的前后置条件, 从而预测 0day 攻击。Wang 等^[3-4]基于漏洞存在及利用的最坏情况假设, 构造 0day 漏洞利用规则, 在生成全局 0day 攻击图的基础上, 提出了 k-0day 安全度量,

依据攻陷目标节点所需 0day 漏洞最小数量评估网络安全性, 该研究虽然实现了 0day 攻击危害性的度量, 但在最坏情况假设下生成的全局攻击图, 规模过大, 难以应用于节点规模较大的网络。而 Albanese 等^[5]则提出了无须事先生成全局攻击图即可评估系统 k-0day 安全性的算法, 提高了 k-0day 安全度量的可行性。然而以上研究均以安全度量为研究重点, 未深入涉及 0day 攻击的推理方法。Yang 等^[6]参照已知漏洞利用规则, 构建提权、DDoS 及信息泄露等多种 0day 攻击模型, 并提出了较为完整的 0day 攻击图生成算法, 但该算法不加约束地假设系统每个设备均存在一个 0day 漏洞, 导致攻击图中存在大量发生概率较低的 0day 攻击。叶子维^[7]等基于已知攻击路径及权限提升原则, 对 0day 攻击条件进行了约束, 排除了部分冗余的 0day 攻击, 然而其关于 0day 攻击的推理依赖前期生成完整的已知攻击路径, 难以应对已知漏洞较少, 不足以构建完整攻击路径的场景。

0day 攻击的实时检测是指基于攻击图等预测结果, 融合实时告警信息, 通过攻击级联, 依据被检测的已知攻击间的“跳跃”状况, 对可能已发生的 0day 攻击进行的推理。Ahmadinejad 等^[8]通过设置攻击级联阈值, 判断出现攻击“跳跃”的路径是否出现未知攻击。Sun^[9-11]等通过构建系统调用依赖图, 结合入侵报警信息, 在系统调用层面实现了 0day 攻击检测, 提高了检测精度。而 Singh 等^[12]则提出了利用并行处理提高检测实时性的分层式 0day 攻击检测架构。然而, 以上攻击实时检测的研究本质上仍需以攻击预测生成的结果作为先验知识实施推理。

综上所述,攻击预测技术是0day攻击检测研究的关键。然而,关于0day攻击预测的研究普遍依靠假设的条件、专家知识构建的攻击模型以及同一攻击路径中攻击的前后依赖关系来应对0day漏洞未知性的影响,这一过程存在3个方面的不足:一是条件假设缺乏有效约束,易导致0day攻击的预测结果规模过大,降低了预测的意义;二是专家知识构建的攻击模型,易受专家主观知识面的制约;三是在已知攻击路径不完整的情况下,预测方法难以适用。针对以上不足,本文提出了基于知识图谱的0day攻击路径预测方法,利用网络防御知识图谱将攻击相关的威胁、资产、脆弱性等数据融合为知识相互关联且覆盖面较广的安全知识库,依据其中整合的漏洞数据、攻击意图等知识,对0day漏洞未知属性的假设条件进行合理约束;其次,利用路径排序算法,以知识图谱中链接攻击者实体与目标实体的关系路径为特征,从更全面的角度对0day攻击展开预测,克服专家知识构造模型的局限性;最后,以历史攻击数据为样本,设计并训练Logistic二元分类器,实施单步攻击预测,从而摆脱对已知攻击路径的依赖,并通过复用Logistic二元分类器输出的单步攻击发生概率,计算攻击路径综合利用率,预测针对目标资产最有可能被攻击者利用的0day攻击路径,从而支持防御决策^[13-14]。

1 预备知识

为使表述清晰准确,对文中相关概念定义如下。

定义1 网络防御知识图谱(CKG, cyber defense knowledge graph)

CKG以三元组(CSO, FACT, T)表示,其中,CSO=(C, R, P)为网络安全本体,C为类集,R为关系类型集,P为属性类型集,FACT为以RDF(resource description framework)三元组格式表示的数据知识的集合,T为CSO中的类与FACT中的实体对象的类型从属关系集合。

定义2 0day漏洞

0day漏洞指未被安全厂商发现,却可能被黑客组织掌握的系统脆弱性的总称。为使研究更具针对性,以下0day漏洞仅指未被安全厂商发现的

技术漏洞。

定义3 0day攻击

0day攻击指攻击者利用掌握的0day漏洞发起的单步攻击,记为 a^0 。相对地,已知攻击 a^k 为攻击者利用已知漏洞发起的单步攻击。

定义4 0day攻击路径zap

zap是指包含0day攻击的一组前后依赖的单步攻击构成的无环攻击序列,表示为 (A, E) ,其中,A为单步攻击集,E为链接单步攻击的有向边集。

定义5 0day攻击图ZAG

ZAG指包含0day攻击的攻击图,表示为 $(A, Priv, L, Prob)$,其中, $A=\{a^0\} \cup \{a^k\}$ 为0day攻击与已知攻击组成的单步攻击集,单步攻击 a 以二元组(host, vul)表示,host为目标设备,vul为利用的脆弱性,Priv为单步攻击的前后置权限集, $L=\{A \times Priv\} \cup \{Priv \times A\}$ 为单步攻击与权限间存在的链接,表示相互间的前后置关系,Prob为单步攻击发生概率的集合。

定义6 关系路径(rp, relation path)^[15]

rp指由知识图谱中的一组关系类型组成的序列,写作 $rp: c_0 \xrightarrow{r_1} c_1 \xrightarrow{r_2} \dots \xrightarrow{r_l} c_l$, $r \in R$ 。其中, $c_i = \text{Domain}(r_{i+1}) = \text{Range}(r_i)$, $c_0 \equiv \text{Domain}(rp)$, $c_l \equiv \text{Range}(rp)$, $l = |rp|$ 表示关系路径长度,为关系路径包含的关系类型总数, $0 \leq i \leq l$ 。

本文中知识图谱与攻击图的关系如下:CKG是攻击预测算法的输入,作为知识库,为攻击预测提供所需的知识;0day攻击图ZAG则是攻击预测结果的图形表示。

关系路径与攻击路径的区别如下:rp用作路径排序算法中logistic regression模型执行攻击预测使用的特征;0day攻击路径zap则是以0day攻击图参照,结合多步攻击发生概率,提取的攻击路径预测结果。

对本文中出现的符号描述如表1所示。

2 网络防御知识图谱

知识图谱是一种利用图模型描述知识并建模万物间关联关系的有效技术方法^[16]。将该技术应用于网络安全领域,构建网络防御知识图谱,可将异构碎片化的网络安全数据整合为格式统一、互相关联的安全知识,为攻击预测提供支持,

表 1 本文主要符号及其描述
Table 1 Symbols and Their Descriptions in This Paper

符号	描述	符号	描述
CKG	网络防御知识图谱	vul	单步攻击利用的漏洞
CSO	网络安全本体	rp	关系路径
zap	0day 攻击路径	c	本体中的类
ZAG	0day 攻击图	r	本体中的关系类型
A	单步攻击集	Domain(r)	关系类型的定义域
a	单步攻击	Range(r)	关系类型的值域
Priv	节点权限集	s	源实体
L	单步攻击与权限间的有向链接集	d	目标实体
E	攻击路径中单步攻击间的有向边集	h	关系路径特征值
Prob	单步攻击发生概率集	H	关系路径特征向量
host	发生攻击的设备	θ	关系路径权重向量

有利于实施针对性防御。以下重点对网络防御知识图谱架构构建以及网络安全本体设计进行详细介绍。

2.1 架构构建

根据定义 1 可知, 网络防御知识图谱可分为模式层与数据层两部分, 其中, 模式层为知识图谱的核心, 以网络安全本体 (CSO, cyber security ontology) 定义了网络防御的基本概念体系, 为数据层知识的建模提供模式定义。数据层则是知识图谱的主体, 是通过知识抽取、知识融合等步骤获取的具体事实, 在模式定义下建模而成的数据知识的集合, 数据知识以 RDF 三元组形式表示为 (subject, predicate, object)。两层关系示例如图 1 所示, 该示例的模式层定义了攻击模式类、脆弱性类以及分别以两者为定义域、值域的关系类型 exploit, 数据层则以此模式建模了数据知识 (CVE-2017-0290, Code Injection, exploit), 表示代码注入这一攻击模式能够利用漏洞 CVE-2017-0290。

根据知识图谱的分层结构, 其构建主要有自顶向下、自下而上两种方法。由于当前网络安全领域概念体系的研究已较为成熟^[17-19], 因此网络防御知识图谱适用于自顶向下的知识图谱构建方法, 即首先构建以网络安全本体为内容的模式层, 再以模式层为基础, 综合多种知识抽取及融合技术, 从异构数据源提取并建模数据知识, 构建数据层。网络防御知识图谱架构构建如图 2 所示。

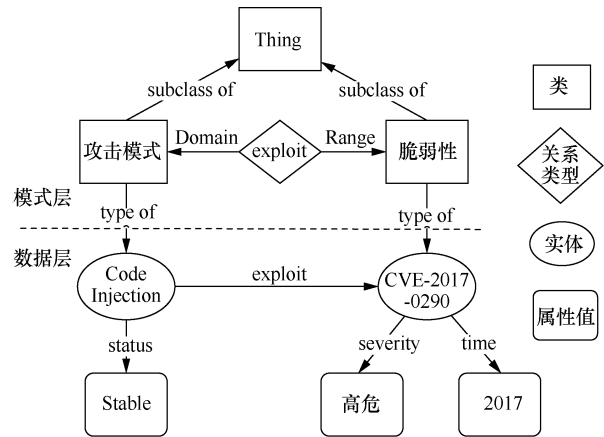


图 1 模式层与数据层关系示例
Figure 1 Relation of pattern layer and data layer

图中绿色、橙色箭头分别代表模式层、数据层构建涉及的步骤, 蓝色箭头则为两者共同涉及的步骤, 箭头序号表示步骤次序。对于自顶而下的知识图谱构建方法, 首先抽取现有网络安全领域本体的研究成果, 利用本体集成的方式构建网络安全本体, 完成模式层构建, 在此基础上, 以本体定义的知识模式为参照进行数据层的知识抽取、融合, 实现数据层构建, 知识抽取过程中, 针对关系数据库提供的结构化数据以及以文本为代表的非结构化数据, 可分别使用 D2R (relational database to RDF)^[20] 工具及深度学习的方法^[21]实现知识抽取。最终, 为更好地支持路径排序算法等图结构算法进行知识推理, 使用图数据库将数据知识存储为以节点、边为元素的属性图格式, 完成知识存储, 由图数据

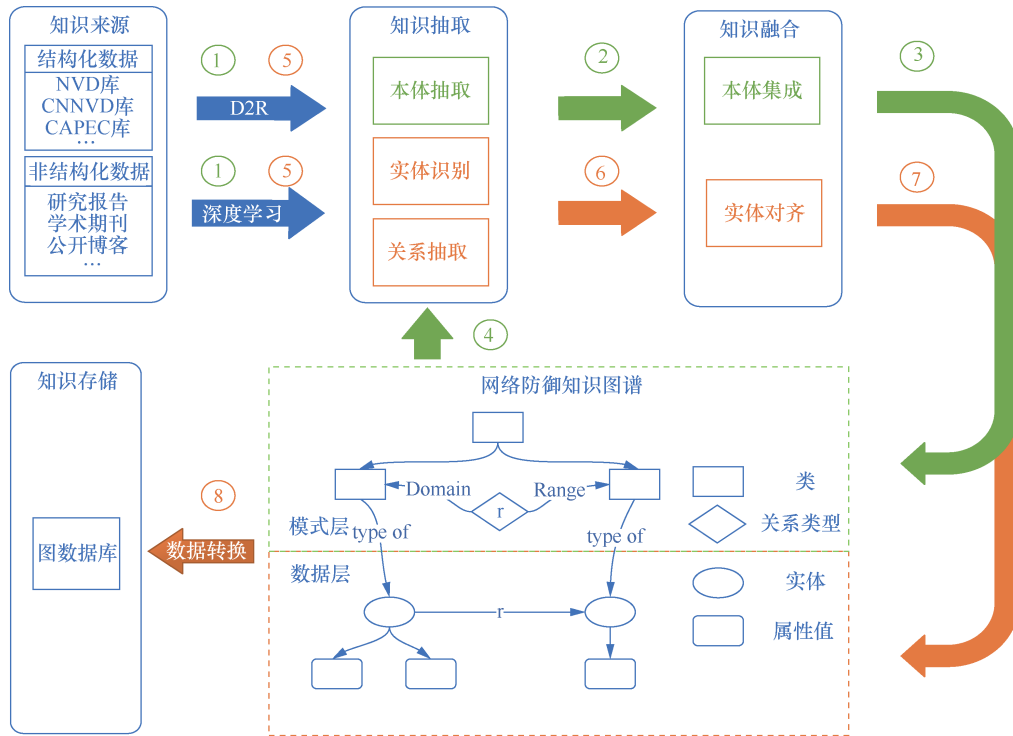


图2 网络防御知识图谱构建架构
Figure 2 Construction framework of CKG

库对外提供统一的知识图谱应用接口。

2.2 本体设计

由自顶而下的知识图谱构建顺序可知，CSO 是决定网络防御知识图谱质量的关键。本文利用本体集成的方式构建 CSO，通过将现有研究较为成熟的网络安全本体融合为一个统一本体，借鉴当前该领域研究成果并实现不同成果间的优势互补。由于司成等^[22]提出的 NSSEKB_O (ontology of network security situation element knowledge base) 从领域本体、应用本体、原子本体 3 个层面对网络安全知识体系进行了系统梳理，具有较好的知识完备性，而谢敏容等^[19]构建的 AFACSDO (asset fragility attack cyber security domain ontology) 复用了多个有代表性的网络安全本体，是网络安全本体研究的最新成果，具有较好的时效性。因此，本文选取以上两项研究成果作为集成对象实施本体集成。

网络安全本体以资产、威胁、脆弱性三大知识模块为基础，融合 NSSEKB_O 及 AFACSDO 的相关概念知识，如图 3 所示。其中，资产模块

以设备类为核心，集成了设备存在的组件，操作设备的职员、设备承载的功能、提供的服务、存储的数据等资产以及设备的权限等类；威胁模块以攻击者类为核心，集成了竞争对手、黑客组织等攻击者类的子类以及攻击者能够运用的攻击模式、攻击模式常用的工具、所属的战术等类；脆弱性模块以脆弱性为核心，集成了脆弱性利用产生的后果，并将脆弱性类细化为已知漏洞、0day 漏洞、人性弱点等子类。为提高图 3 的可视性，图中使用有向边表示不同的关系类型，有向边的起始类及目标类分别为关系类型的定义域与值域。关系类型将同一知识模块中的不同类以及不同知识模块互相关联起来，构成知识体系。

3 0day 攻击路径预测

网络防御知识图谱将攻击这一概念定义为以攻击者类为定义域、设备类为值域的关系类型(图 3 红色虚线)，具体攻击行为的推理问题即转化为知识图谱数据层中攻击者实体与设备实体间攻击关系的预测问题。而路径排序算法 (PRA, path ranking algorithm)^[23]是一种用于知识图谱链路预

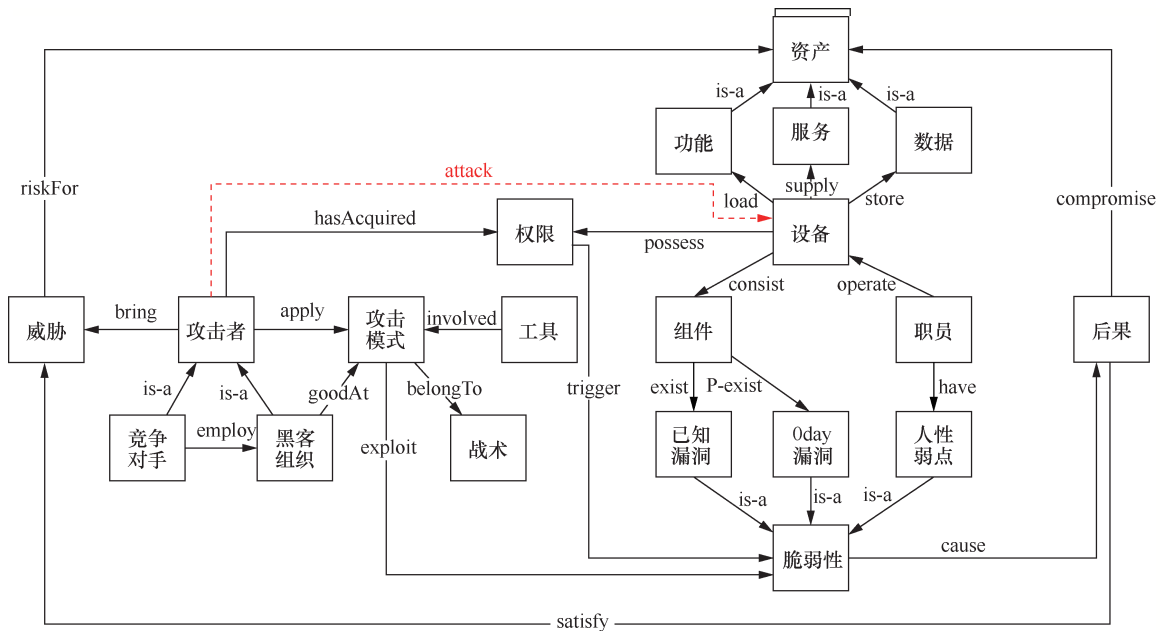


图 3 网络安全本体示意
Figure 3 Cyber security ontology

测的有效方法，其预测结果不仅准确率高，且可解释性强，能够便于防御者在获取预测结果后对攻击成因等知识进行挖掘，因此本文选用 PRA 算法执行单步攻击预测，并实现 0day 攻击图的构建。在此基础上，通过分析比较不同攻击路径的综合利用率，预测攻击者最有可能利用的 0day 攻击路径。

3.1 0day 漏洞未知属性分析

在实施预测前，通过未知属性分析，对缺失的 0day 漏洞属性进行假设并实施约束，这是预测分析的基础。0day 漏洞未知的属性主要包括漏洞存在位置、利用条件以及产生影响等信息。当前相关研究主要通过条件假设的方式对未知的 0day 漏洞信息进行补全，在此过程中，如何对假设的漏洞信息进行合理约束是决定预测结果质量的关键。本文基于知识图谱整合的知识，在提出 0day 漏洞存在性、可用性、危害性假设的基础上，分别采用统计分析、样本训练及意图分析的方式对相关假设进行约束。

3.1.1 0day 漏洞存在性

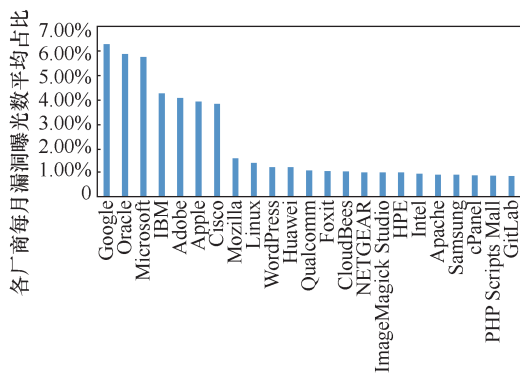
存在性假设：0day 漏洞可能存在于设备的任意组件中。

NVD、CNNVD 等数据库提供的漏洞数据显示，不同组件已曝光的漏洞数量存在明显差别，

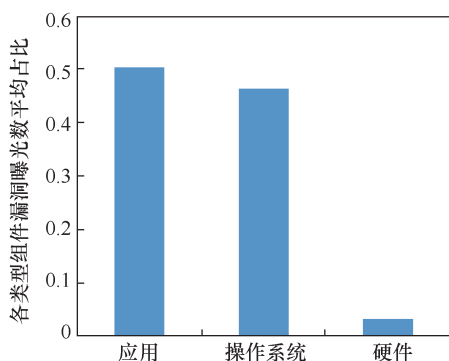
表明漏洞的存在性与作为其载体的组件存在关联。因此对于该假设，可利用组件特征进行约束，将 0day 漏洞视为组件将来可能曝光的漏洞，通过统计分析不同组件的漏洞曝光历史数据，对不同组件存在 0day 漏洞的可能性进行量化。

图 4 为以 CNNVD 公布的 2015 年 7 月以来的漏洞数据为样本，抽取厂商 (vendor)、组件类型 (ctype) 两类组件属性作为特征，按月统计获取的不同属性组件月漏洞曝光数的平均占比，平均占比越高表示具有相应属性的组件相对于其他组件存在 0day 漏洞的可能性越高。如图 4 所示，不同厂商组件的漏洞曝光平均占比由高到低呈现明显的阶梯状下降趋势，据此可将厂商划分为高、中、低 3 类。高占比厂商 (HPV, high production vendor) 包括 Google、Oracle、Microsoft，月平均占比约 6%，中等占比厂商 (MPV, middle production vendor) 包括 IBM、Adobe、Apple、Cisco，月平均占比约 4%，低占比厂商 (LPV, low production vendor) 为 Mozilla 至 GitLab 等，月平均占比约 1%，其余图中未显示的厂商占比相对较低，认为其存在 0day 漏洞的可能性极小，设为 0。而组件类型方面，应用及操作系统类型组件的漏洞曝光数占比均接近 50%，而硬件类型组件远远小于前者，仅占 3%。对以上占比数据进行归

一化处理,构造 0day 漏洞存在概率 p 的赋值矩阵,如表 1 所示。



(a) 不同厂商漏洞曝光数平均占比统计



(b) 不同类型组件漏洞曝光数平均占比统计

图4 漏洞曝光数平均占比统计分析

Figure 4 Statistics of vulnerabilities exposing average proportion

表 1 0day 漏洞存在概率 p 的赋值
Table 1 Existence possibility p of 0day vulnerability

各占比厂商	硬件	操作系统	应用	合计
HPV	0.02	0.26	0.27	0.55
MPV	0.01	0.17	0.18	0.36
LPV	0	0.04	0.05	0.09
合计	0.03	0.47	0.5	1.0

根据存在性假设,在知识图谱中建立 0day 漏洞实体至全部组件实体的关系 “ $P\text{-exist}^{-1}$ ”,记 $\text{indv}(\text{0day})$ 、 $\text{indv}(\text{Cmpt})$ 分别为某 0day 漏洞实体及某组件实体,则两实体间的关系 “ $P\text{-exist}^{-1}$ ” 成立的概率 $\text{prob}(P\text{-exist}^{-1})$ 为: $\text{prob}(P\text{-exist}^{-1}(\text{indv}(\text{0day}), \text{indv}(\text{Cmpt}))) = p(\text{indv}(\text{Cmpt}), \text{vendor}, \text{indv}(\text{Cmpt}), \text{ctype})$

由上述可见,0day 漏洞的存在性假设受到了组件特征的约束,当厂商、类型取值为不同的组合时,0day 漏洞存在于相应组件的概率也不同。

3.1.2 可用性与危害性

可用性假设: 0day 漏洞可能被目标设备上的任意权限触发。

权限在知识图谱中以权限实体的形式存在,不同漏洞的触发需要不同程度的权限,而权限越高,越便于攻击者触发漏洞。在知识图谱中设置关系 “trigger” 的属性 “tri_prob” 表示权限触发 0day 漏洞的概率,依据权限高低进行赋值。关系 “trigger” 包含于攻击者实体与目标设备实体间的关系路径中,因而 “tri_prob” 通过参与路径特征的计算,可体现于样本特征中。在利用攻击样本对 Logistic 二元分类器进行训练的过程中,依据正负样本的区别,可将 0day 漏洞的利用条件约束于样本中的情景。当分类器判定 0day 攻击关系成立时,通过查询知识图谱中攻击者实体在设备实体上已获取的权限实体,可确定本次触发 0day 漏洞所使用的权限。

危害性假设: 0day 漏洞利用产生的危害仅能够满足攻击者实现对目标设备攻击意图的最低需求。

漏洞利用产生的危害在知识图谱中以后果实体的形式表示,同时,知识图谱通过威胁实体,对攻击者的攻击意图进行了整合。由于攻击者利用漏洞发起攻击是以实现攻击意图为目的的,如果漏洞利用产生的后果不足以支持攻击意图的实现,则攻击者没有利用该漏洞的必要,而当后果超出实现攻击意图的需要时,超出的部分对于攻击者而言意义也不大,因此以上假设利用攻击意图约束 0day 漏洞利用产生的危害性是可行且较为合理的。例如,攻击者实体 APT-28 主要从事窃密活动,目标设备实体存储有机密资料,在知识图谱中构建威胁实体 “窃取机密” 作为 APT-28 对该设备的攻击意图,若预测 APT-28 利用 0day 漏洞对该设备发起了攻击,则产生的后果通过威胁实体约束为 “机密性损害”,不再额外涉及完整性、可用性等其他方面的影响。

3.2 路径排序算法

路径排序算法的基本思想是将链接两个实体的关系路径作为特征来预测实体间是否存在某种特定关系。具体而言,该算法依托知识图谱提取实体间的一组关系路径,选择合适的方法计算起始实体沿该路径到达目标实体的概率,记为路径

特征, 以此为输入, 利用训练好的 Logistic Regression 模型综合全部路径特征计算待测关系成立的概率, 并依据阈值判断关系是否成立。

3.2.1 算法示例说明

引用一个简单的 PRA 算法应用实例, 简要说明该算法的操作过程。知识图谱子图如图 5 所示。

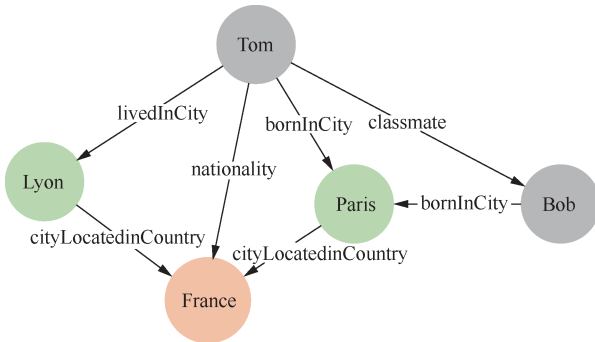


图 5 知识图谱子图的实例
Figure 5 Sub graph of example

需要预测的目标关系为 born In City。第一步, 提取定义域、值域分别为 Person 类及 City 类的关系路径, 并限制路径长度不超过 2; 第二步, 获取目标关系的实例, 正例如 (Tom, Paris), 负例如 (Tom, Lyon); 第三步, 计算关系实例中起止

实体间的路径特征, 构造样本; 第四步, 利用样本数据训练分类器; 最后, 使用训练完成的分类器进行预测。各步骤产生的数据如表 2 所示。

表 2 操作过程中生成的数据
Table 2 Data produced in operation

类别	数据
关系路径	nationality $\rightarrow$ cityLocatedinCountry
	classmate $\rightarrow$ bornInCity
	classmate ⁻¹ $\rightarrow$ bornInCity
	classmate ⁻¹ $\rightarrow$ livedInCity
正例	(Tom, Paris)、(Bob, Paris)
负例	(Tom, Lyon)、(Bob, Lyon)
正样本	((1,1,0,0),1)、((0,0,1,0),1)
负样本	((1,0,0,0),0)、((0,0,0,1),0)

3.2.2 关系路径提取

由定义 6 可知, 关系路径为关系类型的序列, 属于模式层知识, 通过查询网络安全本体, 提取 Domain(rp)=Adversary, Range(rp)=Device, 长度不超过 6 的关系路径 rp, 如表 3 所示。

给定起止实体的情况下, 表 3 中的关系路径反映了实体间在不同方面的关联性, 如路径 rp2 从攻

表 3 关系路径提取
Table 3 Extraction of relationpaths

名称	关系路径
rp1	Adversary <u>hasAcquiredPrivilege</u> <u>possess</u> ⁻¹ Device
rp2	Adversary <u>bringThreatriskFor</u> (Asset <u>is-a</u> ⁻¹ Function) <u>load</u> ⁻¹ Device
rp3	Adversary <u>bringThreatriskFor</u> (Asset <u>is-a</u> ⁻¹ Data) <u>store</u> ⁻¹ Device
rp4	Adversary <u>bringThreatriskFor</u> (Asset <u>is-a</u> ⁻¹ Service) <u>supply</u> ⁻¹ Device
rp5	Adversary <u>applyAttackMode</u> <u>exploit</u> (Vulnerability <u>is-a</u> ⁻¹ Known) <u>exist</u> ⁻¹ Component <u>consist</u> ⁻¹ Device
rp6	Adversary <u>applyAttackMode</u> <u>exploit</u> (Vulnerability <u>is-a</u> ⁻¹ 0day) <u>P-exist</u> ⁻¹ Component <u>consist</u> ⁻¹ Device
rp7	Adversary <u>applyAttackMode</u> <u>exploit</u> (Vulnerability <u>is-a</u> ⁻¹ Human) <u>have</u> ⁻¹ Stuff <u>operate</u> Device
rp8	Adversary <u>applyAttackMode</u> <u>exploit</u> Vulnerability <u>cause</u> Consequence <u>compromise</u> (Asset <u>is-a</u> ⁻¹ Function) <u>load</u> ⁻¹ Device
rp9	Adversary <u>applyAttackMode</u> <u>exploit</u> Vulnerability <u>cause</u> Consequence <u>compromise</u> (Asset <u>is-a</u> ⁻¹ data) <u>store</u> ⁻¹ Device
rp10	Adversary <u>applyAttackMode</u> <u>exploit</u> Vulnerability <u>cause</u> Consequence <u>compromise</u> (Asset <u>is-a</u> ⁻¹ Service) <u>supply</u> ⁻¹ Device
rp11	Adversary <u>hasAcquiredPrivilege</u> <u>trigger</u> (Vulnerability <u>is-a</u> ⁻¹ Known) <u>exist</u> ⁻¹ Component <u>consist</u> ⁻¹ Device
rp12	Adversary <u>hasAcquiredPrivilege</u> <u>trigger</u> (Vulnerability <u>is-a</u> ⁻¹ 0day) <u>P-exist</u> ⁻¹ Component <u>consist</u> ⁻¹ Device
rp13	Adversary <u>hasAcquiredPrivilege</u> <u>trigger</u> (Vulnerability <u>is-a</u> ⁻¹ Human) <u>have</u> ⁻¹ Stuff <u>operate</u> Device
rp14	Adversary <u>applyAttackMode</u> <u>exploit</u> Vulnerability <u>cause</u> Consequence <u>satisfy</u> ThreatriskFor (Asset <u>is-a</u> ⁻¹ Function) <u>load</u> ⁻¹ Device
rp15	Adversary <u>applyAttackMode</u> <u>exploit</u> Vulnerability <u>cause</u> Consequence <u>satisfy</u> ThreatriskFor (Asset <u>is-a</u> ⁻¹ data) <u>store</u> ⁻¹ Device
rp16	Adversary <u>applyAttackMode</u> <u>exploit</u> Vulnerability <u>cause</u> Consequence <u>satisfy</u> ThreatriskFor (Asset <u>is-a</u> ⁻¹ Service) <u>supply</u> ⁻¹ Device

注: r^{-1} 为 r 的逆关系, is-a 表示的父子类关系仅用于标识父类实体的不同种类, 不计入路径长度, 不参与特征计算。

击意图对设备承载功能产生的威胁方面刻画攻击者与目标设备的联系，而 $rp8$ 则从脆弱性对设备承载功能的危害方面刻画了两者间的联系。

3.2.3 路径特征计算

路径排序算法利用随机游走的方法计算路径特征，对于任意关系路径 rp ：

$$c_0 \xrightarrow{r_1} c_1 \xrightarrow{r_2} \dots \xrightarrow{r_l} c_l$$

给定起始实体 $s \in \text{Domain}(rp)$ ，目标实体 $d \in \text{Range}(rp)$ ，则路径特征 $h_{s,rp}(d)$ 计算如下：

当 $rp = \emptyset$ 时，

$$h_{s,rp}(d) = \begin{cases} 1, s = d \\ 0, \text{其他} \end{cases} \quad (1)$$

当 $rp \neq \emptyset$ 时，记 rp' ：

$$c_0 \xrightarrow{r_1} c_1 \xrightarrow{r_2} \dots \xrightarrow{r_{l-1}} c_{l-1}$$

采用递归的方式计算特征，

$$h_{s,rp} = \sum_{d' \in \text{Range}(rp')} h_{s,rp'}(d') \cdot \Pr(d | d'; r_l) \quad (2)$$

其中， $\Pr(d | d'; r_l) = |r_l(d', d)| / |r_l(d', \cdot)|$ 是以实体 d' 为起点沿 r_l 经单步游走到达实体 d 的概率， $|r_l(d', d)|$ 表示实体 d' 与 d 间关系 r_l 的存在性，若关系存在，则 $|r_l(d', d)| = 1$ ，否则为 0。

使用以上方法计算获取的路径特征值本质上决定于游走者（walker）在游走过程中经过的实体在知识图谱中的数量分布，该方法未考虑实体特殊性对关系存在性产生的影响，且对于实体数量分布过于敏感。以关系路径 $\text{Sample: Adversary} \xrightarrow{\text{apply}} \text{AttackMode} \xrightarrow{\text{exploit}} \text{Vulnerability}$ 为例，计算由攻击者实体 a 到脆弱性实体 c 的路径特征，为便于说明，假设知识图谱中 a 能够运用的攻击模式集为 AM ， AM 中任一攻击模式 b 均能够利用的脆弱性 c ，且其能够利用的脆弱性的集合为 V_b ，则根据随机游走算法，第一步游走： a 到达任意一个攻击模式实体 b 的概率 $\Pr(b|a; \text{apply}) = 1/|AM|$ ，第二步游走：由攻击模式集 AM 中的实体游走到脆弱性实体 c ，对于任一实体 $b \in AM$ ， $\Pr(c|b; \text{exploit}) = 1/|V_b|$ ，综合两步游走，路径特征 $h_{a, \text{sample}}(c) = \sum_{b \in AM} \frac{1}{|AM|} \Pr(c|b; \text{exploit}) = \frac{1}{|AM|} \sum_{b \in AM} \frac{1}{|V_b|}$ ，可见，特征计算结果仅与攻击模式实体及脆弱性实体的数量分布相关。对于利用难度不同的脆弱性实体，在相关实体数量分布一致的情况下，具有相同的路径特征值，将此特征值应用于攻击预

测，显然并不合理。且当攻击者 a 新增一种攻击模式，而该模式无法利用脆弱性 c 时，路径特征值受攻击模式数量变化影响，减小为 $\frac{1}{|AM+1|} \sum_{b \in AM} \frac{1}{|V_b|}$ ，

而事实上新增的这一攻击模式由于不能利用脆弱性 c ，不会对路径特征产生影响。鉴于此，为提高路径特征计算的合理性，将实体的特殊性纳入路径特征的计算，并降低路径特征对实体数量分布的敏感性，将式(2)改变如下：

$$h_{s,rp}(d) = \max_{d' \in \text{Range}(rp')} (h_{s,rp'}(d') \cdot \exp(r(d', d))) \quad (3)$$

$\exp(r(d', d))$ 为给定实体 d' 与 d 间关系 r 存在的期望，期望计算如式(4)所示。通过在计算过程中加入关系存在概率 prob ，将实体特殊性产生的影响计入路径特征，并将沿关系路径转移的最大概率作为路径特征，有效消除了无关实体新增对路径特征产生的不合理影响。如出现上节实例的情况时，由于新增攻击模式实体与脆弱性 c 不存在利用关系，经该实体到达脆弱性 c 的概率为 0，不影响最大概率的选取。

$$\exp(r(d', d)) = |r(d', d)| \cdot \text{prob}(r(d', d)) \quad (4)$$

$\text{prob}(r(d', d))$ 取值受实体特殊性影响并需结合实际选择合理的方法进行计算，如 3.1 节对于关系 “P-exist⁻¹” 存在概率的赋值，需通过统计不同属性组件的漏洞曝光历史数据，横向比较数量占比，进行量化计算，此处不再赘述。

3.2.4 攻击分类器构造

路径预测算法利用 Logistic Regression 模型构造二元分类器实施链路预测。在完成路径特征计算的基础上，通过设计得分函数，综合不同的路径特征，计算待测关系成立的得分，得分越高表示关系成立的可能性越高。得分函数如式(5)所示。

$$\text{Score}(s, d) = \sum_{i=1}^n h_{s, \rho_i}(d) \cdot \theta_i \quad (5)$$

其中， θ_i 为关系路径权重，表示相应关系路径对待测关系是否成立的影响程度， n 为路径特征数量。令 $H = (h_{s, \rho_1}(d), h_{s, \rho_2}(d), \dots, h_{s, \rho_n}(d))^T$ ， $\theta = (\theta_1, \theta_2, \dots, \theta_n)^T$ ，则得分函数可写作 $\theta^T H$ ，以其为输入，二元变量 y 表示待测关系是否成立，Logistic 二元分类器构造如下：

$$\Pr(y = 1 | H; \theta) = \frac{1}{1 + e^{-\theta^T H}} \quad (6)$$

上式表示取定参数 θ ，路径特征向量为 H 的情况下，待测关系成立的概率。Logistic 二元分类器以 0.5 为阈值，当 $\text{Pr} > 0.5$ 时，判断关系成立。为取定合适的参数 θ ，需利用样本对分类器进行训练，记样本集为 $D = \{(H_j, y_j)\}$ ， $m = |D|$ 为样本规模，Logistic 分类器的损失函数称为交叉熵，如式(7)所示。

$$O(\theta) = \frac{1}{m} \sum_{j=1}^m \left[y_j \log(1 + e^{-\theta^T \cdot H}) + (1 - y_j) \log(1 + e^{\theta^T \cdot H}) \right] \quad (7)$$

训练过程为求解使损失函数在样本集上取值最小的一组参数作为最优解。训练完成后，将参数最优解代入式(6)，通过输入给定两实体间的关系路径特征 H ，可对实体间待测关系进行预测。

3.3 攻击路径预测

利用给定系统的历史攻击数据构造攻击样本，训练分类器 LC_A 用于预测攻击是否发生，在此基础上，从攻击正样本中区分 0day 攻击、已知攻击，构造 0day 攻击样本，训练分类器 LC_Z 用于判断发生的单步攻击是否为 0day 攻击。完成攻击预测后，利用图数据库的查询功能，以起止实体及关系路径为条件，对攻击利用的脆弱性及前后置条件进行挖掘，构造单步攻击，并生成 0day 攻击图。

依托 0day 攻击图，以攻击者初始权限为起点，目标权限为终点，提取 0day 攻击路径，并通过复用 LC_A 分类器输出的单步攻击发生的概率，计算不同攻击路径的综合利用率，以此为依据，预测攻击者最有可能利用的 0day 攻击路径。综合利用率计算如式(8)所示。

$$\text{Exploit}(\text{zap}) = \prod_{a \in \text{zap}} \text{prob}(a) \quad (8)$$

0day 攻击路径预测算法如算法 1 所示。

算法 1 0day 攻击路径预测算法

输入 网络防御知识图谱 CKG；关系路径集 RP；攻击样本 D_1 ；0day 攻击样本 D_2

输出 0day 攻击图 ZAG=(A, Priv, L, Prob)；0day 攻击路径集 ZAP；综合利用率集 CE；最优攻击路径 opt_zap

1) Initialize A, L, Prob, Priv, Thr, Cons, ZAP, CE = $\emptyset$;

```

2) att=selectAttacker(CKG); //选取攻击者
3) Host←selectHost(CKG); //提取系统设备集
4) Init_Priv←Query_Privilege (att, CKG); //
  查询攻击者初始权限集
5)  $LC_A.\text{fit}(D_1)$ ;  $LC_Z.\text{fit}(D_2)$ ; //分类器训练
6) while 1:
7)  $A_{\text{tmp}}=A$ ;
8) for host in Host:
9) Calculate  $H_{\text{att, host}}$ ; //特征计算
10)  $y=LC_A.\text{predict}(H_{\text{att, host}})$ ;
11) if( $y=0$ )//分类器判断攻击未发生，跳出
  本次循环
12) continue;
13) if( $LC_Z.\text{predict}(H_{\text{att, host}})$ )://判断攻击为
  0day 攻击
14) vul=Query_0day (att, host, CKG, RP); //在
  知识图谱中查询攻击利用的 0day 漏洞实体
15) Thr←Query_Threat(att, host, CKG, RP);
  //查询攻击者对目标设备的攻击意图
16) Create post_priv satisfy Thr; //创建满足攻
  击意图的权限实体作为攻击后置权限
17) else
18) vul=Query_Known (att, host, CKG, RP); //
  在知识图谱中查询攻击利用的已知漏洞实体
19) Cons←Query_Consequence (vul, CKG,
  RP); //查询漏洞利用的后果
20) post_priv=Evaluate(Cons); //根据后果评
  估攻击者获取的权限作为后置权限
21)  $a=\text{Create Attack}(\text{host, vul})$ ; //构造单步攻击
22) if( $a \cap A == a$ )://判断攻击不为新攻击，则跳
  出本次循环
23) continue;
24)  $A \leftarrow a$ ;
25)  $\text{prob}(a)=LC_A.\text{prediction\_prob}(H_{\text{att, host}})$ ; //
  提取攻击发生的概率
26) Prob←prob(a);
27) Priv←post_priv;
28)  $L \leftarrow \text{CreateLink}(a, \text{post\_priv})$ ;
29) pre_priv=Query_Privilege(att, host, CKG,
  RP); //查询攻击的前置权限
30) Priv←pre_priv;
```

```

31)  $L \leftarrow \text{CreateLink}(\text{pre\_priv}, a)$ ;
32)  $\text{CKG} = \text{CKG.update}(\text{post\_priv})$ ; //更新知识图谱中攻击者获取的权限
33) if ( $A_{\text{tmp}} == A$ ) //判断本轮预测未出现新的攻击, 则结束预测
34) break;
35) endwhile
36)  $\text{Goal\_Priv} \leftarrow \text{check}(\text{Priv})$ ; //选取目标权限
37)  $\text{ZAP} \leftarrow \text{AttackPathExtraction}(\text{Init\_Priv}, \text{Goal\_Priv})$ ; //提取攻击路径
38) forzapin ZAP:
39) Calculateexploit(zap);
40)  $\text{CE} \leftarrow \text{exploit}$ ;
41)  $\text{opt\_zap} = \text{argmax}(\text{exploit}(\text{zap}), \text{zap})$ ;
42) Return  $\text{ZAG} = (A, \text{Priv}, L, \text{PR}), \text{ZAP}, \text{CE}, \text{opt\_zap}$ 

```

算法的核心为步骤 13~32, 利用步骤 5 训练完成的分类器逐个预测系统设备可能发生的单步攻击, 并区分已知攻击、0day 攻击。依托网络防御知识图谱对攻击利用的漏洞及前后置权限进行挖掘, 按前后置关系将单步攻击链接为攻击图。由于新的攻击可能使攻击者获取新的权限, 因此步骤 29 对知识图谱 CKG 进行了实时更新。执行完一轮预测后, 若步骤 33 判断无新攻击出现时, 预测结束, 输出 0day 攻击图 ZAG, 并进行攻击路径提取; 否则, 开始新一轮预测。由于图数据库查询的代价与知识图谱规模无关, 仅与遍历范围成正比, 而以上查询均以关系路径为约束条件, 关系路径长度不超过 6, 因此每次查询的开销最大为 $O(6)$ 。假设系统中漏洞数量为 N , 设备数量为 M , 每个漏洞均能导致攻击, 且每一轮只能发现一例新的攻击, 则需执行 N 轮预测, 每一轮包括 M 次预测, 算法时间复杂度为 $O(6MN)$ 。

4 实验

为验证本文方法的有效性, 设计实验环境如图 6 所示。该实验环境由 3 个子网组成, 子网间部署防火墙实现访问控制。其中, DMZ 区域部署的 Web 服务器及邮件服务器分别提供对外应用服务接口及对内邮件服务; 子网 1 为办公区, 部署两台主机及一台文件服务器, 文件服务器存储企

业机密文件; 子网 2 为业务区, 部署应用服务器, 为 Web 服务器提供应用业务支撑。

系统中各实体间权限关系矩阵如表 4 所示, 其中, 纵向实体为权限拥有者, 横向实体为权限载体。系统中脆弱性的分布区分为 10 组训练场景及 1 组实验场景, 分别用于训练分类器及实施预测。其中, 使用 9 组训练场景生成的攻击样本训练分类器, 剩余 1 组用于生成测试集, 测试分类器性能参数; 实验场景则主要用于生成 0day 攻击图、预测 0day 攻击路径, 对比现有研究成果, 检验本文方法的优势。表 5 为训练场景 10 与实验场景的脆弱性分布信息。

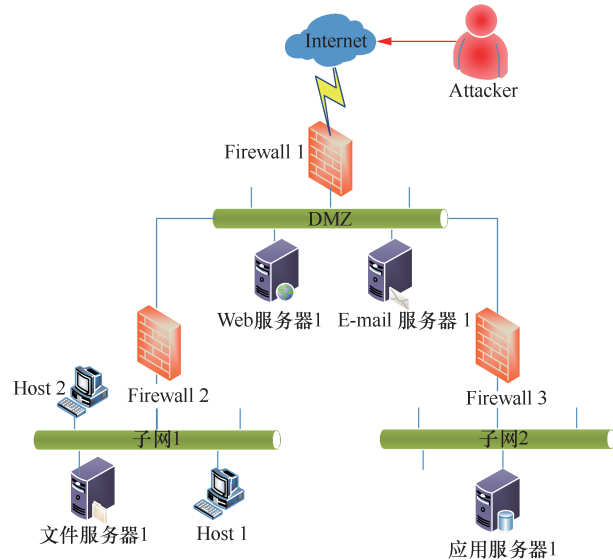


图 6 实验环境
Figure 6 Experiment environment

4.1 图谱生成

利用 Neo4j desktop 1.3.8 图数据库, 以网络安全本体 (图 3) 定义的知识模式为基础, 建模数据层知识, 生成网络防御知识图谱。威胁、资产、脆弱性 3 个知识模块分别以信息工程大学 CTF 战队模拟的黑客组织、实验环境设备以及不同场景中存在的脆弱性为核心构建相应实体, 并从 CAPEC、NVD、CNNVD 等公开数据库抽取实体关系、属性等知识。其中, 威胁知识方面, 攻击者实体命名为“CTF001”, 设定其以窃取文件服务器内信息为主要攻击意图, 对于其余系统设备以制造攻击“跳板”为意图, 其知识图谱表示如图 7 所示。图 7 中粉色模块表示攻击模式实体;

表 4 权限关系矩阵
Table 4 Matrix of privileges between devices

	Attacker	Firewall1	Web server	Email server	Firewall 2	Host 1	Host 2	File Server	Firewall 3	App server
Attacker	—	Access	Access	—	—	—	—	—	—	—
Firewall 1	Access	—	Access	Access	—	—	—	—	—	—
Web server	Access	Access	—	—	—	—	—	—	Access	Access
Email server	—	Access	—	—	Access	Access	—	—	—	—
Firewall 2	—	—	—	Access	—	Access	Access	—	—	—
Host 1	—	—	—	Access	Access	—	—	Access	—	—
Host 2	—	—	—	—	Access	—	—	User	—	—
File Server	—	—	—	—	—	Access	Access	—	—	—
Firewall 3	—	—	Access	—	—	—	—	—	—	Access
App server	—	—	Access	—	—	—	—	—	Access	—

表 5 脆弱性分布情况
Table 5 Distribution of vulnerabilities

设备	训练场景 10 (测试)					实验场景				
	漏洞	E	C	I	A	漏洞	E	C	I	A
firewall_1	CVE-2016-6367	1.8	high	high	high	CVE-2019-1934	2.8	high	high	high
firewall_2	CVE-2017-17152	2.2	none	none	high	CVE-2017-17156	3.9	none	none	high
firewall_3	CVE-2017-17312	3.9	none	none	high	CVE-2014-9136	2.8	high	high	high
Web_Server	CVE-2017-15720	2.8	high	high	high	—	—	—	—	—
Email_Server	CVE-2018-18772	2.2	none	high	high	CVE-2018-18772	2.2	none	high	high
File_Server	CVE-2018-8206	3.9	none	none	high	CVE-2018-8169	2.2	high	high	high
Host_1	CVE-2017-18379	3.9	high	high	high	CVE-2018-12714	2.2	none	high	high
Host_2	CVE-2018-1013	2.2	high	high	high	CVE-2018-8472	2.2	high	none	none
App_Server	CVE-2017-8527	2.8	high	high	high	CVE-2017-8527	2.8	high	high	high

注：E 为漏洞在 CVSS3.1 标准下的可用性评分，C、I、A 分别为漏洞利用对机密性、完整性、可用性的影响程度。



图 7 攻击者实体的知识图谱
Figure7 Attacker entity in knowledge graph

灰色模块表示设备权限实体；棕色模块表示威胁实体，包括对文件服务器的数据窃取（如 `data_stolen_file_server`），对防火墙的功能阻断（如 `function_blocked_firewall_1`）以及对其余设备转发功能的篡取（如 `function_grabbed_web_server`）。脆弱性知识方面，系统中存在的 0day 漏洞实体及相应关系根据 3.1 节的相关假设进行构建。实际实验过程中，选取可用且能满足攻击者对相应设备攻击意图的技术漏洞模拟攻击者掌握的 0day 漏洞。例如，以代码执行漏洞 CVE-2020-14841 模拟 App_Server 的 `weblogic_server` 组件存在的 0day 漏洞，该漏洞仅需远程访问权限即可触发，获取权限越高，越容易触发，符合 0day 漏洞可用性假设，且执行后能够完成目标设备的控制，从而满足攻击者对 App_Server 的攻击意图。

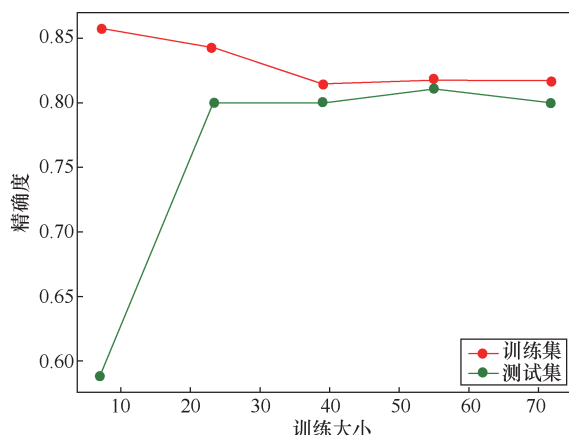
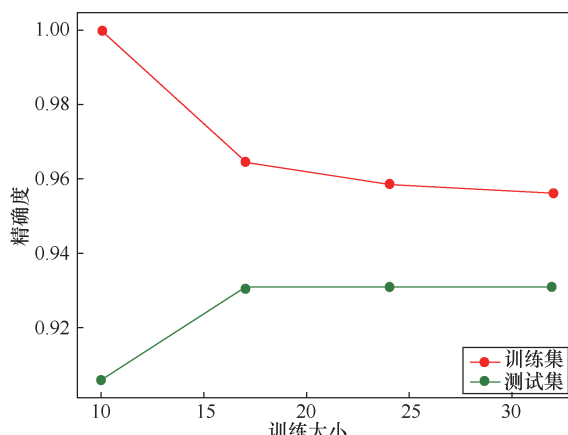
(a) LC_A学习曲线(b) LC_Z学习曲线

图8 分类器学习曲线

Figure 8 Learning curves of classifiers

使用训练场景 10 生成的样本作为测试集，进行分类器测试，对比预测结果与实际攻击情况，获取分类器 LC_A 的精准率为 0.875、召回率为 0.917、调和平均值 F_1 为 0.883。而分类器 LC_Z 对该测试集的预测结果与实际情况相符。由此可见，分类器对 0day 攻击具有较好的识别能力。

4.3 实验结果

利用 4.2 节训练完成的分类器 LC_A 与 LC_Z 对实验场景下的目标系统中可能发生的攻击进行预测，生成 0day 攻击图，如图 9 实线部分所示。

需要注意的是，不同场景下，漏洞种类、数量、分布不同，在计算路径特征时，需根据场景对知识图谱的相应部分进行替换。

4.2 样本训练

以模拟攻击者的 CTF 战队对不同训练场景下的目标系统的攻击数据为基础，依托知识图谱，以其中被攻击成功的设备为目标实体，计算路径特征，构建攻击正样本 $\{(H_i, y=1)\}$ ，以攻击失败及未被选择为攻击目标的设备为目标实体，构建攻击负样本 $\{(H_j, y=0)\}$ 。在 Python3.5 环境中使用模型 `sklearn.linear_model.LogisticRegression` 构造二元分类器，利用训练场景 1~9 生成的攻击样本训练 LC_A。在此基础上，在攻击正样本中，区分 0day 攻击与已知攻击，分别构造 0day 攻击正样本及负样本，训练 LC_Z。使用 5-fold 交叉验证，获取分类器学习曲线，如图 8 所示。

由图 9 可知，预测攻击过程为：攻击者首先利用远程访问权限对 `firewall_1` 发起 0day 攻击，突破访问控制后，获取 E-mail_Server 的远程访问权限，利用其操作系统 CentOS 存在的 CVE-2018-18772 漏洞，发起跨站请求攻击，获取 Host_1 及 `firewall_2` 的访问权限，可分别利用两者存在的 CVE-2018-12714、CVE-2017-17156 漏洞发起代码注入攻击，通过 Host_1 获取 File_server 的访问权限，通过 `firewall_2` 获取 Host_2 的访问权限，此时，可直接利用访问权限对 File_server 发起 0day 攻击，或对 Host_2 发起 0day 攻击，获取 File_Server 的用户权限，从而触

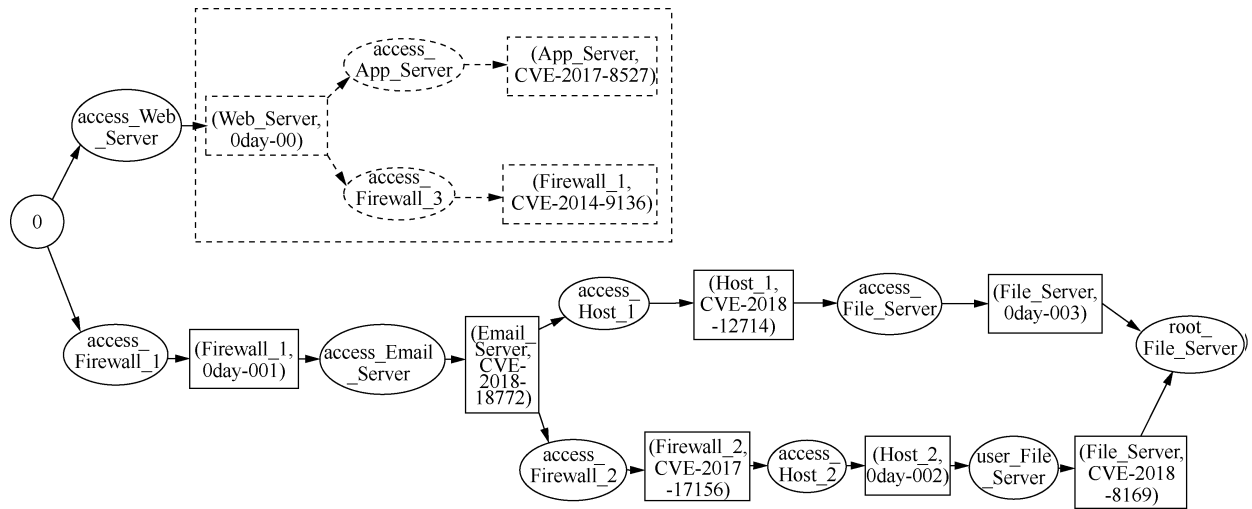


图 9 实验场景 0day 攻击图
Figure 9 0day attack graph in experimental scene

发已知漏洞 CVE-2018-8169。LC_A 输出单步攻击的发生概率如表 6 所示。

表 6 攻击发生概率
Table 6 Attack probability

代号	单步攻击	攻击发生概率
a_1	(Firewall_1, 0day-001)	0.51 654 167
a_2	(Email_Server, CVE-2018-18772)	0.77 063 776
a_3	(Host_1, CVE-2018-12714)	0.78 201 273
a_4	(File_Server, 0day-003)	0.58 126 107
a_5	(Firewall_2, CVE-2017-17156)	0.76 765 838
a_6	(Host_2, 0day-005)	0.77 269 441
a_7	(File_Server, CVE-2018-8169)	0.9 277 641

以获取 root_File_Server 为最终攻击目的, 提取 0day 攻击路径包括: zap1: $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4$ 以及 zap2: $a_1 \rightarrow a_2 \rightarrow a_5 \rightarrow a_6 \rightarrow a_7$, 利用式(8)计算两路径的综合利用率分别为 0.18、0.21。由此可知, 攻击路径 zap2 虽然涉及 5 次攻击, 多于 zap1 的 4 次攻击, 但其综合利用率更高, 更有可能被该攻击者利用。

4.4 实验分析

根据实验结果, 通过对比本文与文献[6-7]提出的方法, 从预测结果的准确性、方法便捷性、适用性、相关知识的全面性以及预测结果可解释性等方面分析本文方法的优势, 以验证本文方法的有效性。具体对比分析如表 7 所示。

表 7 对比分析
Table 7 Comparison analysis

方法	准确性	便捷性	适用性	知识全面性	可解释性
文献[6]提出的方法	较低	较低	较高	较低	不支持
文献[7]提出的方法	较高	较低	较低	较低	不支持
本文方法	较高	较高	较高	较高	支持

准确性方面, 利用文献[6]提出的方法生成的 0day 攻击图除图 9 实线部分外, 还包括虚线所示部分, 并且文献[6]简单的根据 0day 攻击数量预测攻击路径 zap1 更可能成为攻击者选择的攻击路径。而本文通过对 0day 存在性假设的约束 (Web_Server 中存在 0day 漏洞可能性最高的为 apachewebserver 组件, 可能性为 0.05), 并利用训练完成的分类器 LC_A 对 Web_Server 发生攻击的概率进行了计算 (攻击发生概率 0.3), 判定攻击关系不成立, 较为合理地排除了此处的 0day 攻击, 缩减 0day 攻击预测结果规模, 同时根据综合利用率更合理地预测 zap2 为攻击者选择的攻击路径, 提高了预测准确性。

便捷性方面, 文献[6-7]在实施预测前不仅需要收集相关知识, 且需构建专门的 0day 攻击规则作为预测分析的基础, 造成了一定的开销, 而本文依托网络防御知识图谱及知识图谱推理方法实施预测, 无须专门构建 0day 攻击规则, 节省了开销, 提高了方法的便捷性。

适用性方面,文献[7]提出的方法,需依赖较为完整的已知攻击路径实施攻击推理,而本实验环境中,Web_server 不存在已知漏洞且 firewall_1 的已知漏洞 CVE-2019-1934 需用户级权限触发,因而无法在初始权限条件下生成已知攻击路径,文献[7]的方法无法适用于此类场景。

知识全面性方面,本文以当前研究已较为成熟的网络安全领域概念知识为基础,从威胁、资产、脆弱性 3 个方面构建网络防御知识图谱,并提取关系路径作为特征应用于攻击预测,预测过程不仅使用了文献[6-7]等涉及的漏洞存在性、可用性、影响等漏洞知识,也结合了攻击意图、资产类型等知识,使预测分析更加全面,提高了预测结果的合理性。

除以上优势外,本文方法生成的预测结果也具有较好可解释性,以发生在 Host_2 的 0day 攻击(Host_2, 0day-005)为例,Host_2 存在已知漏洞 CVE-2018-8472,不需要特殊权限即可触发,然而,分类器 LC_Z 判断攻击者攻破 firewall_2 获取访问 Host_2 的能力后,最终实施了 0day 攻击。依托知识图谱查询实体“CTF001”与“Host_2”间存在的路径如图 10 所示,依据图中实体及关系,分析攻击者发起 0day 攻击的原因包括两方面:一方面,由图 10 中路径 CTF001 $\xrightarrow{\text{bring}}$ func-

tion_grabbed_Host_2 $\xrightarrow{\text{riskFor}}$ retransmission_Host_2 $\xrightarrow{\text{loaded}}$ Host_2 可知,攻击者以夺取 Host_2 的转发功能为目的,但是 CVE-2018-8472 属性显示,该漏洞仅对目标的机密性产生影响,攻击者无法利用该漏洞实现攻击意图;另一方面,Host_2 的 windows 7 系统属于 Microsoft 推出的操作系统,存在 0day 漏洞的可能性相对较高(表 1 赋值为 0.26),且根据 0day 漏洞危害性假设,0day 攻击能够满足攻击者对目标设备攻击意图。因此,导致攻击者更倾向于使用 0day 攻击,而不是利用已知漏洞。

综合以上 5 个方面可见,本文提出的方法有效可用。

5 结束语

本文针对 0day 漏洞未知性导致的 0day 攻击检测难问题以及现有研究在利用条件假设、攻击前后关联等方法克服未知性影响过程中出现的不足,提出了基于网络防御知识图谱的 0day 攻击路径预测方法。利用当前研究较为成熟的网络安全本体知识,构建了知识覆盖较全面的网络防御知识图谱,将离散的威胁、脆弱性、资产知识整合为高度关联的知识体系,为攻击预测提供了完备的知识支撑。在此基础上,将攻击预测问题转化为链接预测问题,选取预测

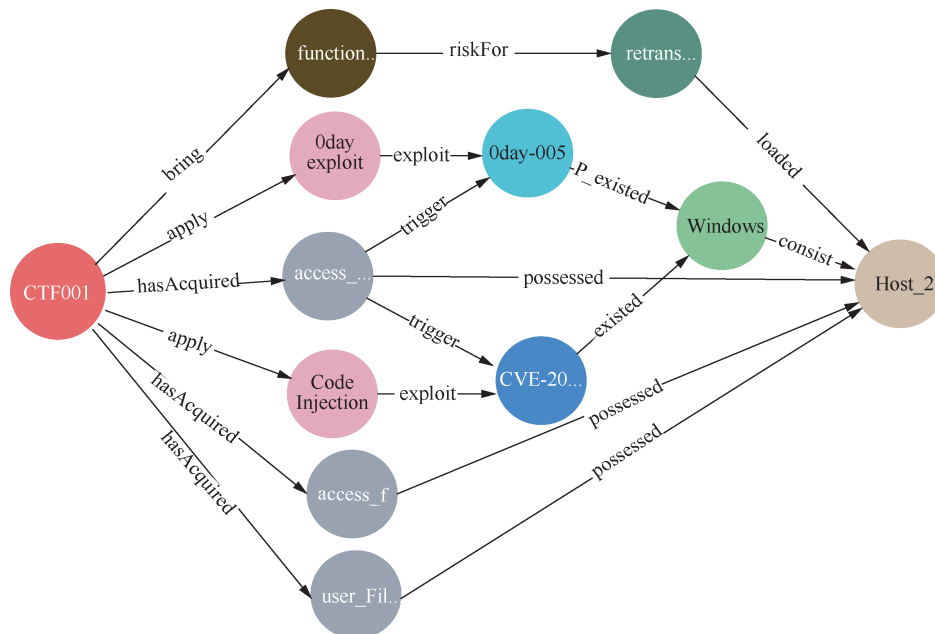


图 10 实体“CTF001”与“Host_2”间路径
Figure 10 Paths between “CTF001” and “Host_2”

精度高且预测结果可解释性强的路径排序算法,提取攻击者实体与目标设备实体存在的关系路径作为特征,更全面地对攻击进行了预测,有效克服了 0day 漏洞未知性以及专家知识片面性的影响,提高了预测准确性,并对预测结果的可解释性提供了支持。

下一步,在持续优化网络防御知识图谱的基础上,将进一步拓展知识模块,提高攻击预测的准确性,并对同时存在多攻击者情况下,基于知识图谱的网络攻击者溯源问题展开探索。

参考文献:

- [1] MCQUEEN M A, MCQUEEN T A, BOYER W F, et al. Empirical estimates and observations of 0day vulnerabilities[C]//2009 42nd Hawaii International Conference on System Sciences. 2009.
- [2] ZHANG S, CARAGEA D, OU X. An empirical study on using the national vulnerability database to predict software vulnerabilities[C]//Database and Expert Systems Applications-22nd International Conference, 2011.
- [3] WANG L, JAJODIA S, SINGHAL A, et al. k-zero day safety: measuring the security risk of networks against unknown attacks[J]. Lecture Notes in Computer Science, 2010, 11(1): 573-587.
- [4] WANG L, JAJODIA S, SINGHAL A, et al. k-zero day safety: a network security metric for measuring the risk of unknown vulnerabilities[J]. IEEE Transactions on Dependable & Secure Computing, 2014, 11(1): 30-44.
- [5] ALBANESE M, JAJODIA S, SINGHAL A, et al. An efficient framework for evaluating the risk of zero-day vulnerabilities[C]// International Conference on E-Business and Telecommunications. 2013.
- [6] YANG Y U, XIA C H, XIAO-YUN H U, et al. An augmented 0-day attack graph generation method[J]. DEStech Transaction on Computer Science and Engineering, 2016(aice-ncs).
- [7] 叶子维. 基于漏洞与攻击图的网络脆弱性分析关键技术研究[D]. 郑州: 信息工程大学, 2019.
YE Z W. Research on key technology of network weakness analysis based on vulnerability and attack graph[D]. Zhengzhou: Information Engineering University, 2019.
- [8] AHMADINEJAD S H, JALILI S, ABADI M. A hybrid model for correlating alerts of known and unknown attack scenarios and updating attack graphs[J]. Computer Networks, 2011, 55(9): 2221-2240.
- [9] SUN X, DAI J, LIU P, et al. Towards probabilistic identification of zero-day attack paths[C]//2016 IEEE Conference on Communications and Network Security (CNS). 2016.
- [10] SUN X, DAI J, LIU P, et al. Using bayesian networks for probabilistic identification of zero-day attack paths[J]. IEEE Transactions on Information Forensics and Security, 2018: 1-10.
- [11] SUN X, DAI J, LIU P, et al. Using bayesian networks to fuse intrusion evidences and detect zero-day attack paths[M]// Network Security Metrics. 2017.
- [12] SINGH U K, JOSHI C, KANELLOPOULOS D. A framework for zero-day vulnerabilities detection and prioritization[J]. Journal of Information Security and Applications, 2019, 46: 164-172.
- [13] 杨峻楠, 张红旗, 张传富. 基于不完全信息随机博弈的防御决策方法[J]. 网络与信息安全学报, 2018, 4(8): 12-20.
- YANG J N, ZHANG H Q, ZHANG C F. Defense decision-making method based on incomplete information stochastic game[J]. Chinese Journal of Network and Information Security, 2018, 4(8): 12-20.
- [14] 冷强, 杨英杰, 常德显, 等. 面向网络实时对抗的动态防御决策方法[J]. 网络与信息安全学报, 2019, 5(6): 58-66.
LENG Q, YAGN Y J, CHANG D X, et al. Dynamic defense decision method for network real-time confrontation[J]. Chinese Journal of Network and Information Security, 2019, 5(6): 58-66.
- [15] LAO N, COHEN W W. Relational retrieval using a combination of path-constrained random walks[J]. Machine Learning, 2010, 81(1): 53-67.
- [16] AMIT S. Introducing the knowledge graph: things, not strings[EB].
- [17] LI A. A practical approach to constructing a knowledge graph for cybersecurity[J]. Engineering, 2018, 4(1): 53-60.
- [18] 尚怀军. 面向漏洞库的网安知识库构建技术的研究与实现[D]. 长沙: 国防科技大学, 2018.
SHANG H J. Research and implementation of technique on construction of cybersecurity knowledge base for vulnerability database[D]. Chagnsha: National University of Defense Technology, 2018.
- [19] 谢敏容. 网络防御知识图谱构建技术研究及实现[D]. 成都: 电子科技大学, 2020.
XIE M R. Research and implementation of cybersecurity knowledge graph construction technology[D]. Chengdu: University of Electronic Science and Technology of China, 2020.
- [20] SAHOO S S, HALB W, HELLAMNN S, et al. A survey of current approaches for mapping of relational databases to RDF[R]. W3C RDB2RDF Incubator Group Report, 2009, 1: 113-130.
- [21] LIN Y, SHEN S, LIU Z, et al. Neural relation extraction with selective attention over instances[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016.
- [22] 司成. 基于本体的网络威胁态势推演与评估技术研究[D]. 郑州: 信息工程大学, 2015.
SI C. Research on network threats situation deduction and evaluation technology based on ontology[D]. Zhengzhou: Information Engineering University, 2015.
- [23] LAO N, MITCHELL T M, COHEN W W. Random walk inference and learning in a large scale knowledge base[C]//Conference on Empirical Methods in Natural Language Processing. DBLP, 2011.

[作者简介]



孙澄(1991-),男,江苏常州人,信息工程大学硕士生,主要研究方向为APT检测跟踪。

胡浩(1989-),男,安徽池州人,博士,信息工程大学讲师,主要研究方向为网络态势感知。

杨英杰(1971-),男,河南郑州人,博士,信息工程大学教授,主要研究方向为信息安全。

张红旗(1962-),男,河北遵化人,博士,信息工程大学教授、博士生导师,主要研究方向为网络安全、移动目标防御、等级保护和信息安全管理。