Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
Cybersecurity—Article

一种构建网络安全知识图谱的实用方法

贾焰，亓玉璐，尚怀军，江荣，李爱平 *

School of Computer Science, National University of Defense Technology, Changsha 410073, China

ARTICLE INFO

Article history:
Received 10 December 2017
Revised 21 December 2017
Accepted 7 January 2018
Available online 9 February 2018

关键词
网络安全
知识图谱
知识推演

摘要

网络攻击的形式复杂多变，检测和预测这些动态类型的攻击是一项充满挑战的任务。在当前的许多领域中，对于知识图谱的研究已经非常成熟。目前，有学者提出将知识图谱的概念与网络安全结合在一起来构建网络安全知识库，这是一件非常有意义的工作。基于这种理念，本文提出了一个构建网络安全知识图谱的方法和基于五元组模型的推演规则。本文使用机器学习的方法来抽取实体，然后构建本体，从而构建网络安全知识库。在构建网络安全知识库的过程中，使用 Stanford NER 来训练提取器，然后利用提取器抽取所需的相关信息。本文提出的推演规则是基于五元组模型的，新的属性是通过计算公式推导得到的，新的关系是基于路径排序算法，同样是通过计算公式推导得到的。

1. 引言

目前，在网络安全领域中，已经构建了相关的网络安全知识库。例如，常见的漏洞数据库，数据库中的每个漏洞都有一个由MITRE公司定义的统一ID，除了漏洞ID，数据库中也包含威胁等级、威胁类型等漏洞的其他信息。过程知识库提供一些常见过程的基本信息，许多知名的防病毒供应商已经建立了巨大的有关病毒的签名库。此外，知识和信息的首要来源是互联网[1]，在互联网中包含大量与网络安全相关的内容，如安全博客、黑客论坛、安全公告等。充分利用来自各种知识库和网站的网络安全相关信息，并将所有这些与安全相关的知识按照一定的规则关联在一起，对入侵检测和网络安全态势感知的相关工作有很重要的意义。

本文的主要工作分为两部分：第一部分论述了构建网络安全知识库的三个步骤，并提出了一个构建网络安全知识库的框架：首先，通过收集和分析结构化数据和非结构化数据来获取相关信息；其次，根据已获得的相关信息构建网络安全本体；最后，完成网络安全知识图谱的构建。第二部分讨论网络安全知识的推演。提出了一个基于网络安全知识库的五元组模型，基于这个模型，介绍如何推演出新的知识，包括新的属性和新的关系。

本文提出的基于网络安全知识库的五元组模型[2]包含以下五个要素：概念、实例、关系、属性和规则。该模型为本体构建提供了基础。文中使用机器学习的方法来抽取与网络安全相关的实体。本文使用Stanford命名实体识别器（NER）来抽取与网络安全相关的实体，

* Corresponding author.
E-mail address: liaiping@nudt.edu.cn (A. Li).

2095-8099/
英文原文: Engineering 2018, 4(1): 53–60
引用本文: Yan Jia, Yulu Qi, Huaijun Shang, Rong Jiang, Aiping Li. A Practical Approach to Constructing a Knowledge Graph for Cybersecurity. *Engineering*, https://doi.org/10.1016/j.eng.2018.01.004

并使用Stanford NER base来训练网络安全领域的抽取模型。为了验证Stanford NER中useGazettes特征的影响，我们构建了三种不同的模型。实验结果表明，useGazettes在网络安全领域中训练NER是非常重要的。在知识推演部分，知识推演包括属性推演和关系推演。对于属性推导，可以使用属性值预测公式来得到新的属性；对于关系推理，基于路径排序算法并使用关系推理预测公式来得到实例之间新的关系。

本文的内容结构如下：第2部分讨论相关工作，第3部分详细介绍本文提出的框架，第4部分提出了一个知识推演方案，第5部分对全文进行总结并对未来工作提出建议。

2. 相关工作

2.1. 本体构建

来自马里兰大学的Undercoffer等[3]完成了一项重要的工作，他们开发了一个模拟攻击和相关实体的本体，但是这个本体只针对攻击，具有一定的局限性。为了表示与网络安全领域相关的概念和实体，Joshi等[4]根据Undercoffer提出的本体论提出了网络安全本体论。他们将本体扩展为可以捕捉国家漏洞数据库（NVD）模式结构和安全漏洞概念的模型关系，该本体包含11种实体类型（如漏洞、产品、方法、后果等）。More等[5]同样也扩展了Undercoffer提出的本体论，他们为推理逻辑添加了规则，该本体包含三个基本类别：方法、后果和目标。

此外，MITRE公司也在研究如何为网络安全领域开发本体[6,7]，MITRE公司为网络安全领域内的特定领域创建了多个标准和数据集，他们有着丰富的数据库。基于Undercoffer和MITER的努力，Iannacone等[8]提出了网络安全知识库的本体论，这个本体代表了一个迭代设计过程的结果，该过程旨在创建知识的表示，可以有效地将来自不同数据源的数据合并到网络安全领域中，该本体包含15个实体类型和115个属性。

2.2. 信息抽取

信息抽取技术对于知识图谱的构建至关重要，因此引起学者们的广泛关注。目前，主要的知识抽取方法有以下两种。

第一种方法主要是基于知识工程。这种方法在很大程度上依赖于提取规则，但可以处理特定领域中的信息提取问题。早期的大部分信息提取系统都是基于提取规则的，这种方法的缺点是需要与领域相关的专业人士和语言学家参与系统的开发。但是这种方法的优点是信息抽取的精确率较高，因此，目前许多信息提取系统还是基于知识工程的。

Pinkston等[9]描述了一个系统，系统中将抽取的攻击信息定义为本体。这项工作是基于对4000多个入侵的修订和该攻击系统所遵循的策略完成的，引入了攻击手段和攻击效果类，同时，采用基于规则的方法构建语义网。Rehman和Mustafa[10]介绍了一个从漏洞文本描述中提取信息的系统。在这个系统中，使用一个基本的频率逆文档频率（TFIDF）评分，从常见漏洞和暴露（CVE）中提取信息。Lowis和Accorsi[11]提出了另一种面向服务架构（SOA）漏洞的分类方法：在CVE上使用简单的字符串匹配将数据集分类为不同的类别，并不是使用机器学习技术对CVE进行分类，该工作重点是针对SOA漏洞的分类。这种基于规则的方法优点是分类的精度高，但是，对于没有明显规则的术语，必须使用其他方法进行抽取。

第二个主要方法是基于机器学习的方法。该方法的基本步骤：使用大量训练数据、训练信息提取模型；训练好的信息提取模型可以用来提取所需要的相关信息。该方法不需要预先由专业人员定义规则，但是需要足够的训练数据才能取得好的结果。

Lal等[12]提出了一个系统，该系统可以从非结构化文本中识别网络攻击和软件漏洞等相关实体，同时，训练了一个NER来识别网络安全的实体。Mulwad等[13]开发了一个框架，主要用于从Web文本中检测和提取有关漏洞和攻击的信息。文中训练了SVM分类器，用来识别潜在的漏洞描述，分类器使用标准的unigram词袋向量模型。一旦发现可能的漏洞描述，它们就使用标准的命名实体识别工具（如OpenCalais）来提取与网络安全相关的实体和概念。以上两项研究均采用机器学习的方法，从非结构化文本自动提取安全相关信息。如果没有足够的训练数据，这种方法无法准确识别安全相关实体，对于特定领域的信息抽取，则采用基于规则和机器学习的方法。

2.3. 网络安全知识库

（1）漏洞数据库。现有的比较丰富的漏洞库有中国信息安全漏洞数据库[14]和美国国家漏洞库[15]。这些数据库收集了各种漏洞。漏洞信息包括漏洞名称、漏洞

描述、漏洞优先级、破坏方法、相应的特征和其他信息等。目前，中国和美国建立的漏洞数据库都遵循常用的命名标准，使不同的数据库中的漏洞信息可以按照统一标准使用。该标准极大地促进了漏洞信息的共享。

（2）攻击规则库。知识库收集已经存在的攻击的信息。这些信息包括攻击名称、攻击类型、协议、攻击特征、攻击描述、严重性和其他属性。其中，Snort攻击规则库是一个相对完善的攻击规则库，库中的每个规则都作为一行存储在一个文件中。

2.4. 知识推理

基于知识的推理大致可以分为基于符号的推理和基于统计的推理。在人工智能研究中，基于符号的推理通常基于经典逻辑（一阶谓词逻辑或命题逻辑）或经典逻辑变体（如默认逻辑）。基于符号的推理不仅可以使用规则推断现有知识图中实体之间的新关系，还可以对知识图进行逻辑冲突检测。基于统计的推理方法通常利用机器学习的方法，通过统计规则，从知识图谱中学习新的实体之间的关系。

类型推理在知识图谱中的目的是学习知识图谱中实例和概念之间的关系。SDType方法[16]使用由三元组或谓词连接的属性的统计分布来预测实例的类型，该此方法可用于任何单个数据源知识图，但不能跨数据集使用。Tipalo[17]是一个用于自动输入DBpedia实体的工具，而链接超义数据集（linked hypernyms dataset，LHD）[18]使用唯一的抽象数据通过特定模式提取实例类型，这种方法依赖于结构化文本数据，不能扩展到其他存储库。

模式推理法主要可以分为基于归纳逻辑程序（ILP）的方法和基于关联规则挖掘（ARM）的方法。基于ILP的方法结合了机器学习和逻辑编程技术，使用户能够从实例和背景知识中得出逻辑结论。Lehmann等[19]提出了一种定义公理的方法，使用向下定义的算子的概念来学习描述逻辑，从最一般的概念（顶级概念）开始，并使用启发式搜索使概念不断专业化以达成概念的定义。Hellmanndengren[20]进一步扩展了这种方法，该方法可以用来处理DBpedia等大规模语义数据，这些方法都在DL-Learner[21]中实现。Völker和Niepert[22]介绍了一种统计方法，用于在知识图谱中生成概念关系，这些知识图谱通过SPARQL查询访问信息以构建事务表，然后使用基于ARM的方法来抽取事务表中的相关的概念关系。在相关的后续工作中，Fleischhacker和Völker[23]使用负相关规则提取技术来研究非概念关系的概念，并且Völker等[24]给出了丰富的实验结果。

3. 框架设计

图1描述了构建网络安全知识库的方法。该框架主要涉及三部分：数据源、信息抽取及本体构建、网络安全知识图谱的生成。

数据源可以分为结构化数据和非结构化数据。本文提出了一种信息抽取方法，该方法基于规则的方法和机器学习的方法。框架底部显示的本体为信息抽取奠定了基础。

如图1所示，知识以图形的形式存储。知识图谱是Google于2012年首次提出的概念[25]。它是一个语义网络，以图形的形式存储实体和实体之间的关系。知识图谱的优点是显而易见的：其相关查询的效率高于传统存储方法，而且由于存储型的灵活，更新起来也很方便。由于网络安全的垂直知识的构建必须考虑知识的深度和整体层次结构，因此，采用了自上而下的方法：首先构建了网络安全本体论，基于构建的本体论，网络安全信息从结构化和非结构化数据中提取。以下讨论着重于本体构建和知识提取。

3.1. 网络安全本体构建

本文提出了一个基于网络安全知识库的五元组模型，该模型包含：概念、实例、关系、属性和规则。网络安全知识库的架构如图2所示。

该架构如图2所示，包含三个本体：资产、漏洞和攻击。图3表示网络安全本体。如图3所示，网络安全本体包含五个实体类型。

（1）漏洞：漏洞数据库中的每条记录都对应于漏洞类型的一个实例。每个漏洞都有其唯一的CVE ID。

（2）资产：在本文中，资产包括软件和操作系统（OS）。

（3）软件：这是资产的一个子类（如Adobe Reader）。

（4）操作系统：这是资产的一个子类（如Ubuntu 14.04）。

（5）攻击：大多数攻击可被视为针对某个漏洞的入侵，如攻击的过程可能是一个漏洞利用的过程。

3.2. 网络安全相关实体的提取：基于机器学习的方法

条件随机场（CRF）是基于统计序列识别和分割的

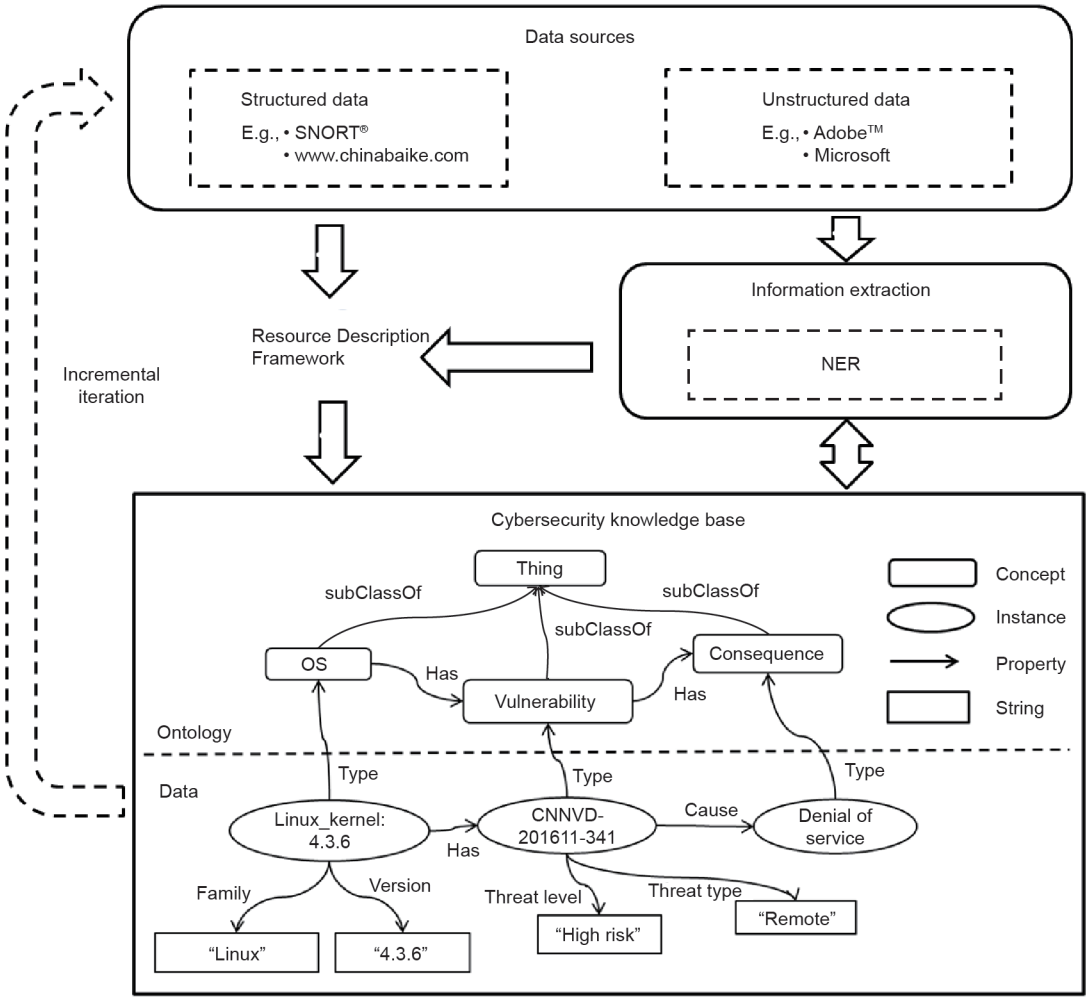

图1. 构建网络安全知识图谱的框架。

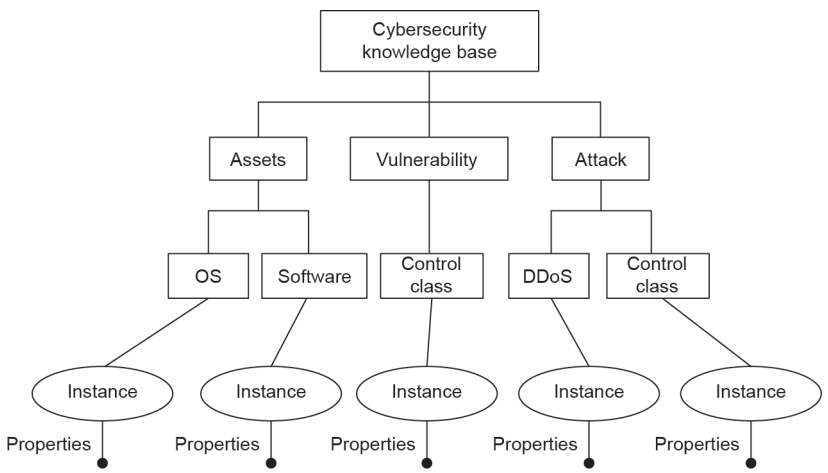

图2. 网络安全知识库架构。

无向图模型。该模型的主要思想来自最大熵模型，其最简单的形式是线性CRF，其中模型中的节点形成线性结构，而线性CRF对应于有限状态机，它非常适合标记线性数据序列。

命名实体识别问题可以被定义为序列的标注问题，即观察到的词是否属于预定义的特征集合。条件随机场是序列标注的概率模型。它没有独立的假设，可以任意选择特征，并且全局规范化所有特征，并获得全局最优

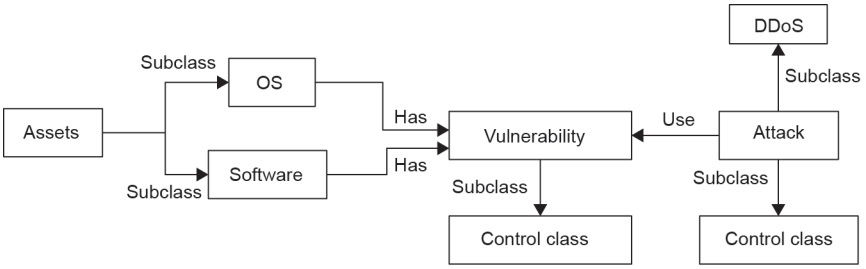

图3. 网络安全本体。

解。它保留了条件概率框架的优点，如最大熵Markov模型。它也解决了标记偏差的问题。因此，条件随机场模型适用于命名实体识别。线性CRF目前是命名实体识别的最佳方法[26]。它是概率分布$P(y|x)$模型，其中，x是观测序列，y是标记序列。$P(y|x)$通过以下公式计算：

$$P(y|x) \propto \prod_{j=1}^{N} \exp \sum_{i=0}^{M} \lambda_i f_i(y_{j-1}, y_j, x_j) \quad (1)$$

式中，N是词语的数目；M是特征的数目。而f_i是一个二值函数，具体的取值如下公式所示：

$$f_i(y_{j-1}, y_j, x_j) = \begin{cases} 1, & \text{if } y_{j-1} \text{ is OS, } y_j \text{ is OS,} \\ & \text{and } x_j \text{ is XP} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Stanford NER [27]提供了线性链条件随机场（CRF）序列模型的一般实现，因此，它也被称为CRF分类器。在本文中，使用Stanford NER来提取与网络安全相关的实体。Stanford NER提供的特征有很多选项。本文使用Stanford NER base来训练提取模型，是因为我们的目标也是在网络安全领域训练NER。

构建模型时选择特征非常重要。在这里，应该选择一个能够更好地识别网络安全相关实体的特征。特征的良好组合是训练良好提取模型的关键。Stanford NER提供超过70个可用于训练模型的特征[28]。确定合适的特征不是一件容易的工作，因为这些斯坦福对这些特征没有太多的解释文档。现有的特征选择算法对于我们的工作不会有太大的帮助。我们必须自己分析已有的特征，选择我们认为对训练模型有用的特征。然后通过实验验证我们的想法。经过多次实验，我们确定了一个用于训练命名识别器的特征集。基于这个特征集训练NER取得了较好的识别效果。我们决定用来训练命名识别器的特征集如下。

（1）UseNGrams：利用n-gram取特征，即词的子串。

（2）MaxNGramLeng：这个特征的值类型为整型。如果这个特征的值为正，则大于该值的n-gram将不会在模型中使用。在本文中，我们将maxNGramLeng的值设置为6。

（3）UsePrev：这可以给我们提供<previous word，class of previous word>的特征，并与其他选项一起启用，如<previous tag，class>。这导致基于当前单词与一对<previous word，class of previous word >之间的关系的特征。当有连续的词属于同一个类时，这个特征将是非常有用的。

（4）UseNext：和UsePrev特征非常相似。

（5）UseWordPairs：这个特征基于两个词对——<Previous word，current word，class> 和 <current word，next word，class>。

（6）UseTaggySequences：这是一个重要的特征，它使用类的序列而不是单词的集合，而是使用第一、第二和第三顺序类和标签序列作为交互特征。

（7）UseGazettes：如果为真，则由下一个名为“gazette”的特征将把文件指出为实体字典。

（8）Gazette：该值可以是一个或多个文件名（以逗号、分号或空格分隔的名称）。如果从这些文件加载公开的实体词典，每行应该是一个实体类名称，后跟一个空格，后面再跟上一个实体。

（9）CleanGazette：如果这个值为真，则仅当全部词在字典中被匹配时，此特征才会触发。如果在字典中有一个词“Windows 7”，那么整个词应该在字典中进行匹配。

（10）SloppyGazette：如果这个值为真，词和字典中的词局部匹配也能触发这个特征，如“Windows ”可以和“windows 7”进行匹配。

本文使用了字典相关的特征，Stanford NER提供两个字典相关特征的具体实现。实验证明了gazette和cleanGazette选项是非常好的选择，因为它们提高了软件和操作系统类的识别准确性。

为了使用这个特征，我们从漏洞数据库中的

influence platform字段汇总了相关的信息，并构建了实体字典。字典中的第一列是实体类型，第二列对应于特定实体。

4. 知识推演

4.1. 数据源

漏洞包括可以收集的现有漏洞，攻击包括目前最流行的攻击技术。其中，漏洞的来源有CVE、NVD、SecurityFocus、CXSECURITY、Secunia、中国国家漏洞数据库（CNVD）、CNNVD和安全内容自动化协议中国社区（SCAP）。攻击的数据来源主要包括两类：一类来自信息安全网站，其中包括Pediy BBS、Freebuf、Kafan BBS和开放Web应用安全项目（OWASP）；另一类来自企业自建信息响应中心，包括360安全响应中心（360SRC）和阿里巴巴安全响应中心（ASRC）。

4.2. 原理分析

K表示知识图谱，K=<concept，instance，relation，properties，rule>，其中：

（1）Concept={concept$_i$|i=1，…，n}。概念是抽象本体的集合，如操作系统、软件、攻击等。

（2）Instance={instance$_i$|i=1，…，m}，实例是具体例子的集合，如Win 7、Adobe Reader、DDoS等。

（3）Properties={<instance$_i$, Pro$_{ij}$,value$_j$>}，属性是实例属性值的集合。

（4）Relation=<concept$_i$,R_{cc},concept$_j$>|<concept$_i$,R_{ci},instance$_j$>|<instance$_i$,R_{ii},instance$_j$>，表示实例之间的关系，如such as subClassOf、instanceOf、is a (ISA)等。

（5）Rule={rule|rule=<instance$_i$, newR_{ij}, instance$_j$>|<concept$_i$, newR_{ij},instance$_j$>|<instance$_i$,Pro$_{ij}$, newValue$_j$>,based on K}，规则用来推演新的属性值和新的关系。

本文提出的知识图谱模型中最重要的部分是规则，这些规则可以用来推演出新的关系和新的属性值。以下重点讨论如何推演出新的属性值和新的关系。

属性推演是使用实例和实例存在的属性来推演出新的属性。例如，姚明的一个属性是他的生日，一旦知道了当前年份和姚明的生日，就可以得到他的年龄。另外一个例子，如果记录显示主机被扫描的次数，则可以使用简单统计来获得总扫描数。

关系推演是使用实例之间存在的关系来推演出实例之间的新关系。例如，关系1（李四，出生地，北京）、关系2（李四，住所，上海）、关系3（北京，属于，中国）、关系4（上海，属于，中国）、关系5（李四，国籍，中国）、关系6（李四，同学，张三）和关系7（张三，出生地，北京），根据以上7个关系，就可以得到一种新的关系：关系8（张三，国籍，中国）。

4.3. 推演结果

4.3.1. 属性推演

如图4所示，图中有三个实例：N_i、N_j和N_l，每一个实例对应一对（key，values）值。

属性由（节点、键、值）对表示。属性值Value$_{ik}$的预测公式如下：

$$\mathrm{Value}_{ik}=\sum_{j=1}^{m}\lambda_j\cdot f_{ij}(\mathrm{key}_j,\mathrm{value}_j)+\sum_{t=1}^{l}\sigma_t\cdot\sum_{j=1}^{m}\lambda_j\cdot f_{ij}(\mathrm{key}_j+\mathrm{value}_j) \quad (3)$$

对于实例N_i，通过计算下述公式可以得到新的属性，如图5所示。

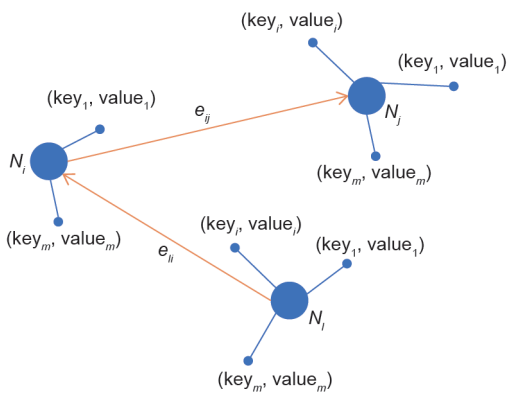

图4. 实例的属性。

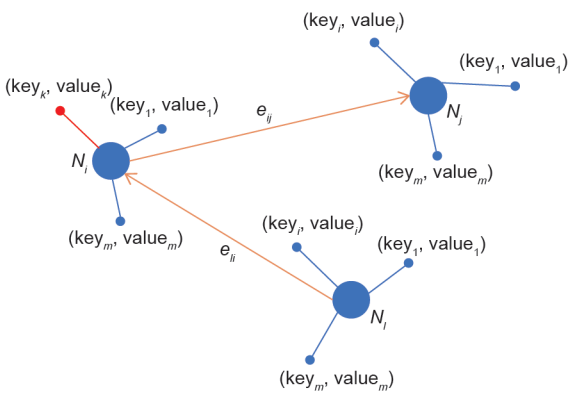

图5. 产生一个新的属性值。N_i的红色部分是新的属性值。

4.3.2. 关系推演

常用推理方法有三种：基于嵌入的技术、基于低维向量表示和路径排序算法。在本文中，我们选择使用路径排序算法。路径排序的基本思想是使用连接两个实体的路径作为特征来预测两个实体之间的关系。对于给定的关系，通过使用路径排序算法，我们可以确定两个实体之间是否存在这种关系。实例N_i、N_j和N_k之间的属性值和关系如图6所示。

关系推理的预测公式如下：

$$\text{Score}(l,j)=\sum_{\pi\in Q}\text{Path}[e_l,e_j;\text{length}(\pi)\leqslant n]\cdot\omega_\pi \tag{4}$$

式中，π为所有从l到j的可达路径，$\text{length}(\pi)\leqslant n$。如果$\text{Score}(l,j)\geqslant\tau$，$\tau$为阈值，则$e_{lj}$成立；否则不成立。通过路径排序算法可以得到新的关系，如图7所示。

4.4. 评估标准

在信息检索和提取系统中，有两个主要的评估指标，包括精确率和召回率。有时，为了全面评估系统的性能，通常计算精确率和召回率的调和平均值。这就是我们通常所说的F-Measure。在本文中，我们使用F-Measure的特殊形式F_1值。精确率、召回率和F_1值由真正、假正和假负定义。定义如下：

（1）真正（TP）：将正类预测为正类数。

（2）假正（FP）：将负类预测为正类数。

（3）假负（FN）：将正类预测为负类数。

精确率（Precision）由以下公式给出：

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{5}$$

召回率（Recall）由以下公式给出：

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{6}$$

F_1值由以下公式给出：

$$F_1=2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}} \tag{7}$$

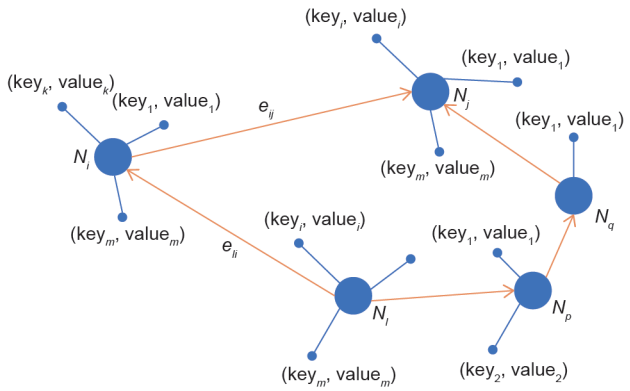

图6. 实例之间的关系。

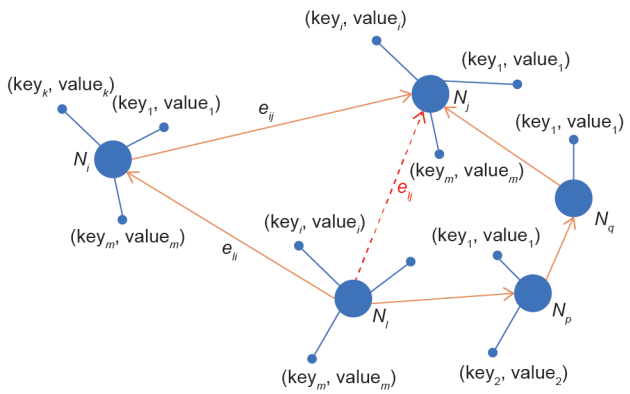

图7. 实例之间新的关系。图中带箭头的红色虚线是新的关系 e_{ij}。

4.5. 实验结果

我们将来自4个数据源的注释数据合并为一个数据集。为了验证useGazettes的影响，我们建立了三个模型（NER_1、NER_2和NER_3）。NER_1没有使用useGazettes作为它的特征，而NER_2使用useGazettes并选择了cleanGazette选项。NER_3也使用了useGazettes，但是它选择了sloppyGazette选项。然后，采用了10倍交叉验证的方法来评估这些模型，将数据分成10个数据块，将十分之九的块用作训练数据，其余用作测试数据。表1显示了三种模型的精确率、召回率和F_1度量的平均值。

在开始时，我们选择了没有使用useGazettes特征来训练NER_1。NER1的平均识别结果如表1所示，该表显示结果的精度相对较高。对于F_1度量，软件和漏洞的识别率相近，且高于其他任何实体类型，也就是说，软件和漏洞的识别在整体识别效果方面取得了良好的表现。

表 1 NER_1识别结果

Entity	Precision	Recall	F_1
Software	0.700	0.795	0.745
OS	0.779	0.691	0.732
Vulnerability	0.805	0.689	0.743
Attack	0.822	0.597	0.692
Total	0.739	0.735	0.737

然后，我们使用了包含useGazettes的特征，并选择了cleanGazette的选项来训练NER_2。NER_3是根据包含useGazettes和sloppyGazette两个选项的特征进行训练的。平均识别结果如表2所示。

表2 NER_2和NER_3识别结果

Model	Entity	Precision	Recall	F_1
NER2 (cleanGazette)	Software	0.809	0.838	0.823
	OS	0.752	0.875	0.809
	Vulnerability	0.753	0.632	0.688
	Attack	0.884	0.559	0.685
	Total	0.789	0.799	0.794
NER3 (sloppyGazette)	Software	0.877	0.838	0.857
	OS	0.832	0.904	0.866
	Vulnerability	0.775	0.632	0.696
	Attack	0.875	0.538	0.667
	Total	0.852	0.805	0.828

如表2所示，在NER_2的识别结果中，对软件的识别取得了良好的整体表现。对于NER_3，OS的识别实现了高F_1值。就软件和操作系统的认可而言，NER_3的整体性能比NER_2好。这个结果表明sloppyGazette选项有助于识别与网络安全相关的实体。NER_2和NER_3的后果和平均值的F_1测量值仍然很低，均小于70%。图8给出了这三种模型之间的直观比较。

如图8所示，就软件和OS的识别精度而言，包括召回率和F_1度量，NER_2和NER_3都高于NER_1。结果证实了useGazette在网络安全领域训练一个NER的重要性。对于攻击效果和攻击手段，实体字典很难构建。没有字典，识别精度不算太高。我们可以采用上文中介绍的基于规则的方法来提取。

5. 总结与展望

本文构建了一个基于漏洞的网络安全本体论，并提出构建网络安全知识库的方法。Stanford NER被用来训练一个提取器来抽取与网络安全相关的实体；但是，识别精度需要进一步提高。另外，可以使用推演规则获得新的实体属性和实体之间的新关系。未来最重要的工作是丰富网络安全知识库和推理规则，并将其应用到入侵检测和态势感知的相关研究中。

致谢

感谢国家重点研究发展计划（No.2016YFB0800802，No.2016YFB0800804，No.2017YFB0802204）和国家自然科学基金（No.61472433，No.61672020，No.U163215）的支持。

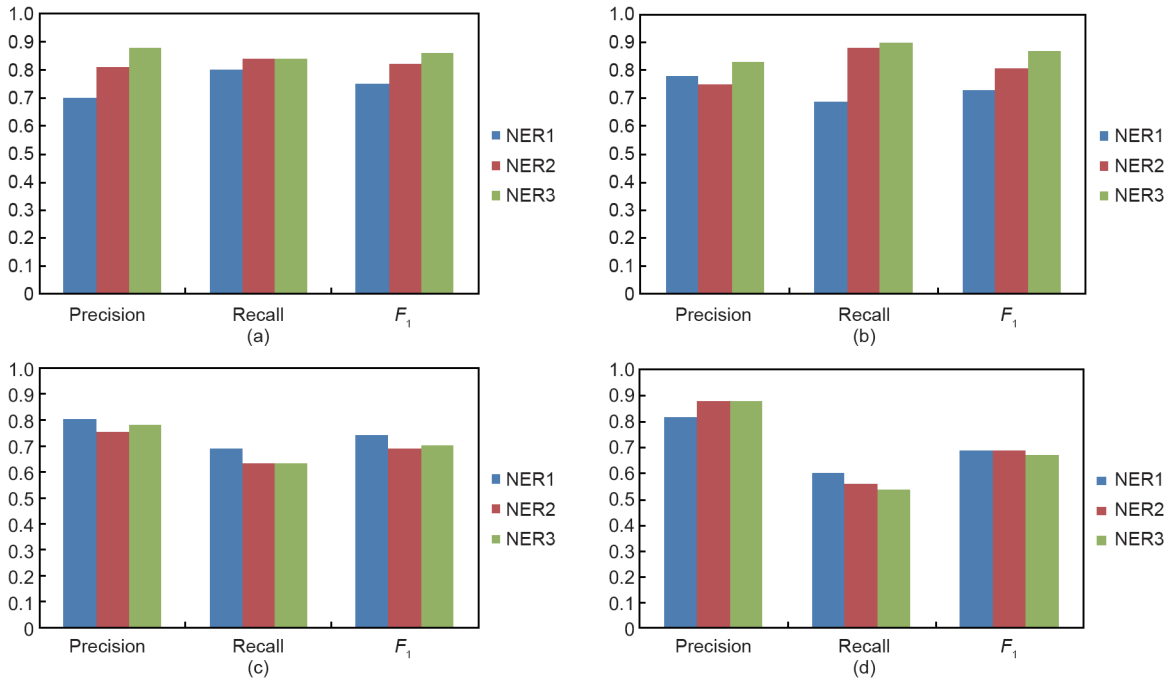

图8. 每一个实例类型的识别结果。

Compliance with ethics guidelines

Yan Jia, Yulu Qi, Huaijun Shang, Rong Jiang, and Aiping Li declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Zhu J, Zhang J, Zhang C, Wu Q, Jia Y, Zhou B, et al. CHRS: Cold start recommendation across multiple heterogeneous information networks. IEEE Access 2017;5:15283–99.

[2] Zhu X, Huang J, Zhou B, Li A, Jia Y. Real-time personalized twitter search based on semantic expansion and quality model. Neurocomputing 2017;254:13–21.

[3] Undercoffer J, Joshi A, Pinkston J. Modeling computer attacks: An ontology for intrusion detection. In: Vigna G, Jonsson E, Kruegel C, editors. RAID 2003: Recent advances in intrusion detection. Berlin: Springer; 2003. p. 113–35.

[4] Joshi A, Lal R, Finin T, Joshi A. Extracting cybersecurity related linked data from text. In: Proceedings of the 7th IEEE international conference on semantic computing. Los Alamitos: IEEE Computer Society Press; 2013. p. 252–9.

[5] More S, Matthews M, Joshi A, Finin T. A knowledge-based approach to intrusion detection modeling. In: Proceedings of 2012 IEEE symposium on security and privacy workshops. Los Alamitos: IEEE Computer Society Press; 2012. p. 75–81.

[6] Obrst L, Chase P, Markeloff R. Developing an ontology of the cybersecurity domain. CEUR Workshop Proc 2012;966:49–56.

[7] Parmelee MC. Toward an ontology architecture for cyber-security standards. CEUR Workshop Proc 2010;713:116–23.

[8] Iannacone M, Bohn S, Nakamura G, Gerth J, Huffer K, Bridges R, et al. Developing an ontology for cybersecurity knowledge graphs. In: Proceedings of the 10th annual cyber and information security research conference. New York: ACM, Inc.; 2015.

[9] Pinkston J, Undercoffer J, Joshi A, Finin T. A target-centric ontology for intrusion detection. In: Proceedings of the IJCAI-03 workshop on ontologies and distributed systems, Aug 9–15, 2003, Acapulco, Mexico; 2003. p. 47–58.

[10] Rehman S, Mustafa K. Software design level vulnerability classification model. Int J Comput Sci Secur 2012;6(4):238–55.

[11] Lowis L, Accorsi R. On a classification approach for SOA vulnerabilities. In: Proceedings of the 33rd annual IEEE international computer software and applications conference. Los Alamitos: IEEE Computer Society Press; 2009. p. 439–44.

[12] Lal R. Information extraction of cybersecurity related terms and concepts from unstructured text [dissertation]. College Park: University of Maryland; 2013.

[13] Mulwad V, Li W, Joshi A, Finin T, Viswanathan K. Extracting information about security vulnerabilities from web text. In: Hübner JF, Petit JM, Suzuki E, editors. Proceedings of 2011 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology—workshops. Los Alamitos: IEEE Computer Society Press; 2011. p. 257–60.

[14] CNNVD.org.cn [Internet]. Beijing: China Information Technology Security Evaluation Center; [cited 2017 Jul 25]. Available from: http://www.cnnvd.org.cn/. Chinese.

[15] NVD.nist.gov [Internet]. Gaithersburg: National Institute of Standards and Technology; [cited 2017 Jul 25]. Available from: https://nvd.nist.gov/.

[16] Paulheim H, Bizer C. Type inference on noisy RDF data. In: Alani H, Kagal L, Fokoue A, Groth P, Biemann C, Parreira JX, et al., editors. The semantic web—ISWC 2013: Proceedings of the 12th international semantic web conference. Berlin: Springer; 2013. p. 510–25.

[17] Paulheim H, Bizer C. Type inference on noisy RDF data. In: Cudré-Mauroux P, Heflin J, Sirin E, Tudorache T, Euzenat J, Hauswirth M, et al., editors. The semantic web—ISWC 2012: Proceedings of the 11th international semantic web conference. Berlin: Springer; 2012. p. 65–81.

[18] Kliegr T. Linked hypernyms: Enriching DBpedia with targeted hypernym discovery. J Web Semant 2015;31:59–69.

[19] Lehmann J, Auer S, Bühmann L, Tramp S. Class expression learning for ontology engineering. J Web Semant 2011;9(1):71–81.

[20] Hellmann S, Lehmann J, Auer S. Learning of OWL class descriptions on very large knowledge bases. Int J Semant Web Inf Syst 2009;5(2):25–48.

[21] Lehmann J. DL-learner: Learning concepts in description logics. J Mach Learn Res 2009;10(11):2639–42.

[22] Völker J, Niepert M. Statistical schema induction. In: Antoniou G, Grobelnik M, Simperl E, Parsia B, Plexousakis D, De Leenheer P, et al., editors. The semantic web: Research and applications: Proceedings of the 8th extended semantic web conference. Berlin: Springer; 2011. p. 124–38.

[23] Fleischhacker D, Völker J. Inductive learning of disjointness axioms. In: Meersman R, Dillon T, Herrero P, Kumar A, Reichert M, Qing L, et al., editors. On the move to meaningful internet systems: OTM 2011: Proceedings of confederated international conferences: CoopIS, DOA-SVI, and ODBASE 2011. Berlin: Springer; 2011. p. 680–97.

[24] Völker J, Fleischhacker D, Stuckenschmidt H. Automatic acquisition of class disjointness. J Web Semant 2015;35(Pt 2):124–39.

[25] Singhal A. Introducing the knowledge graph: Things, not strings [Internet]. [updated 2012 May 16; cited 2017 Jul 25]. Available from: https://googleblog.blogspot.com/2012/05/introducing-knowledge-graphthings-not.html.

[26] Lin D, Wu X. Phrase clustering for discriminative learning. In: Proceedings of the 47th annual meeting of the association for computational linguistics and the 4th international joint conference on natural language processing of the AFNLP. Singapore: Suntec; 2009. p. 1030–8.

[27] Finkel JR, Grenager T, Manning C. Incorporating non-local information into information extraction systems by Gibbs sampling. In: Knight K, Ng HT, Oflazer K, editors. Proceedings of the 43rd annual meeting of the association for computational linguistics. Stroudsburg: Association for Computational Linguistics; 2005. p. 363–70.

[28] NERFeatureFactory [Internet]. Stanford: Stanford NLP Group; [updated 2013 Jun 26; cited 2017 Jul 25]. Available from: http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/NERFeatureFactory.html.