文章编号:1671-4598(2024)03-0063-08 DOI:10.16526/j.cnki.11-4762/tp.2024.03.010 中图分类号:TP311 文献标识码:A

基于知识图谱的网络安全漏洞智能检测系统设计

杜艺帆[1]，丛红艳[2]
(1. 西北大学 现代学院，西安 710130；2. 西安工程大学 新媒体艺术学院，西安 710048)

摘要：网络安全漏洞智能检测需要依赖大量的真实数据来进行分析，冗余数据与异常数据的存在会导致检测准确性下降；为保障网络系统稳定运行，提出基于知识图谱的网络安全漏洞智能检测系统设计研究；从结构、逻辑模型以及运行模式3个方面设计网络安全漏洞检测器，实现网络安全漏洞智能检测系统硬件设计；系统软件设计通过网络爬虫采集安全漏洞数据，去除冗余数据与异常数据，根据属性信息识别安全漏洞实体，获取安全漏洞属性信息关系，以此为基础，定义安全漏洞知识图谱表示形式，设计安全漏洞知识图谱结构，从而实现安全漏洞知识图谱的构建与可视化；以上述网络设计结果为依据构建网络安全漏洞智能检测整体架构，制定网络安全漏洞智能检测具体流程，从而获取最终网络安全漏洞智能检测结果；实验结果表明，在不同实验工况背景条件下，设计系统应用后的网络安全漏洞漏检率最小值为1.23%，网络安全漏洞检测F1值最大值为9.50，网络安全漏洞检测响应时间最小值为1 ms，证实了设计系统的安全漏洞检测性能更佳。

关键词：网络安全；智能化；漏洞挖掘；知识图谱；漏洞检测

Design of Intelligent Detection System for Network Security Vulnerabilities Based on Knowledge Graph

DU Yifan[1], CONG Hongyan[2]
(1. School of Modern, Northwest University, Xi'an 710130, China;
2. School of New Media Art, Xi'an University of Technology, Xi'an 710048, China)

Abstract: The intelligent detection of network security vulnerabilities relies on a large amount of real data for analysis, and redundant and abnormal data can lead to a decrease in detection accuracy. In order to ensure the stable operation of network systems, a network security vulnerability intelligent detection system design based on knowledge graph is proposed. The network security vulnerability detector from three aspects of the structure, logical model, and operation mode is designed to achieve the hardware design of the intelligent network security vulnerability detection system. The system software design collects security vulnerability data through web crawlers, removes redundant data and abnormal data, identifies security vulnerability entities according to attribute information, and obtains security vulnerability attribute information relationships. Based on this, it defines the representation form of the security vulnerability knowledge graph, designs the security vulnerability knowledge graph structure, and the construction and visualization of security vulnerability knowledge graph are realized; Based on the above network design results, an overall architecture for intelligent detection of the network security vulnerabilities is constructed to develop the specific process for the intelligent detection of the network security vulnerabilities, and obtain the final intelligent detection results of the network security vulnerabilities. The experimental results show that under different experimental conditions, the minimum network security vulnerability detection rate of the designed system after application is 1.23%, the maximum F1 value of the network security vulnerability detection is 9.50, and the minimum response time of the network security vulnerability detection is 1 ms, confirming that the designed system has a optimal security vulnerability detection performance.

Keywords: network security; intelligence; vulnerability mining; knowledge graph; vulnerability detection

0 引言

网络的飞速发展为人们生产与生活带来了极大的便利，但与此同时网络病毒传播速度也随之加快，网络安全问题日益严重。安全漏洞是造成网络安全问题的关键所在，其主要来源于网络系统脆弱性。随着网络的不断发展与普及，其在人们生产与生活中占据的地位逐渐上升，网络病毒也随之快速传播，网络安全正在经受着前所未有的威胁。若是网络安全防御措施不足，就会受到非法侵入，致使网络

收稿日期:2023-08-02； **修回日期**:2023-09-18。
基金项目:陕西省教育厅2022年度一般专项科研计划项目(22JK0193)。
作者简介:杜艺帆(1988-),女,硕士研究生,助教。
丛红艳(1976-),女,博士生,教授。
引用格式:杜艺帆,丛红艳.基于知识图谱的网络安全漏洞智能检测系统设计[J].计算机测量与控制,2024,32(3):63-70.

关键信息被篡改、偷窃等，严重会造成网络系统的瘫痪，造成较大的经济损失，威胁国家与大众的信息与财产安全。由此可见，如何保障网络安全具有至关重要的现实意义。通过调查研究发现，目前制约网络发展的最大问题就是安全问题，由于网络本质上是“无政府”的公用平台，任何人均可以自由登陆，使得网络安全保障具备较多的不确定因素（漏洞），为黑客提供了可乘之机。网络安全问题主要来源于系统的脆弱性，主要表现在管理脆弱性、技术脆弱性与系统脆弱性。其中，管理脆弱性主要发生在网络系统安全策略制定、实施、配置与控制过程中；技术脆弱性主要发生在硬件与软件设计过程中；系统脆弱性主要发生在安全防护设备运行过程中。由于上述网络系统脆弱性的存在，使得网络具有较多的安全漏洞，使得别有用心的人可以通过安全漏洞在未授权背景下访问或者破坏网络系统，对网络安全造成极大的威胁。任何网络安全问题均是由安全漏洞引起的，对其进行精准检测是提升网络安全的根本手段。

网络安全漏洞检测是一个动态的过程，并且其难度会随着网络覆盖范围的扩大而增加。相较于发达国家来看，中国对于网络安全漏洞检测的研究较晚，但也取得了一定的研究成果。文献［1］在感知网络整体安全态势的基础上，应用黑盒遗传算法进行相应的模糊测试，选取适当的目标函数与测试参数，测试停止后输出结果即为网络安全漏洞检测结果；文献［2］应用数据预处理模块与协同分析模块对网络安全漏洞信息进行预处理与分析，以此为基础，利用 N-gram 算法匹配漏洞信息与已知的漏洞特征，从而实现网络安全漏洞的检测；文献［3］使用被动分簇算法明确簇首与网关节点，利用 AFL 模糊检测工具过采样安全漏洞样本，结合前向反馈网络和支持向量机构建安全漏洞判别模型，将待检测网络运行数据代入到判别模型中，输出结果即为网络安全漏洞检测结果。上述安全漏洞检测系统虽然能够实现安全漏洞检测功能，但是由于应用手段的自身缺陷，均存在着安全漏洞检测效果较差的问题，无法满足网络系统的发展需求，故提出基于知识图谱的网络安全漏洞智能检测系统设计研究。现有安全漏洞数据库具有信息单一、数据分散、数据结构各异等缺点，这是影响安全漏洞检测效果的关键因素。知识图谱的出现可以有效解决上述问题，其能够根据海量的安全漏洞信息构建安全漏洞知识图谱，对安全漏洞信息进行聚合分析，挖掘安全漏洞关联信息，可以为安全漏洞检测提供更多的信息支撑，从而提升安全漏洞检测整体性能。

1 网络安全漏洞智能检测系统硬件设计

作为网络安全漏洞智能检测系统的关键硬件，检测器主要由管理器、检测单元、通信器等部件构成，为了提升网络安全漏洞检测的精准度，对检测器结构、逻辑模型与运行模式进行合理、科学地设计，具体设计过程如下所示。

1.1 网络安全漏洞检测器结构设计

网络安全漏洞检测器结构如图 1 所示。

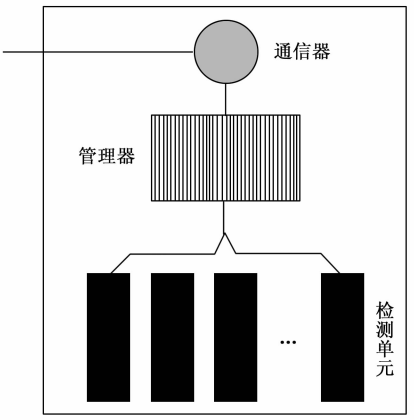

图 1 网络安全漏洞检测器结构示意图

如图 1 所示，在设计的网络安全漏洞检测器结构中，通信器主要承担着安全漏洞数据、漏洞报表、控制指令、网络邮件等的传递任务，与用户、服务器、控制器等进行直接连接，可以实时将安全漏洞检测结果传输给用户与服务器，也可以实时将用户控制指令传输给控制器，保障设计系统的通信顺畅[4]。通信器上述功能主要是在 BSD Socket 的支撑下实现的，还需要遵循一定的数据格式与传输规则，其基础运作原理为 Socket API 函数，具体调用方案需要根据实际情况来制定[5]。

管理器主要作用于检测单元，决定着安全漏洞检测任务执行过程中检测单元如何调度，是检测器中的核心部件。管理器功能实现的关键是配置文件，其中记录了检测单元信息、系统访问权限信息等。当网络系统安全漏洞检测单元增加时，配置文件中也需进行相应的记录。另外，管理器与通信器、检测单元均是直接连接的，用于接收用户反馈的控制指令与检测单元的安全漏洞信息。当管理器接收到通信器传输的控制指令时，先对控制指令进行解译，再以此为基础制定检测单元的控制动作[6]。当管理器接收到检测单元传输的安全漏洞信息时，不需要对其进行解译与分析，只需要将其直接转发给通信器即可。

检测单元是网络安全漏洞检测器的基石，是实现网络安全漏洞智能检测功能的程序实体。标准情况下，一个检测单元对应着一种网络安全漏洞的检测，检测单元之间保持着相互独立的关系。若是存在新的安全漏洞，则应该采用 PERL 语言对新的检测单元进行编制与添加。单一检测单元主要由注册部分、检测部分与卸载部分构成，其管理难度较低，只需要在安装过程中向管理器配置文件进行备份即可。

上述过程完成了网络安全漏洞检测器结构的设计，并对构成部件进行了详细地描述与介绍，为检测器功能的实现奠定基础。

1.2 网络安全漏洞检测器逻辑模型设计

逻辑模型是网络安全漏洞检测器功能实现的主要依据，故此节在用户、网络系统、漏洞检测等多方需求背景下，设计网络安全漏洞检测器逻辑模型，具体如图 2 所示。

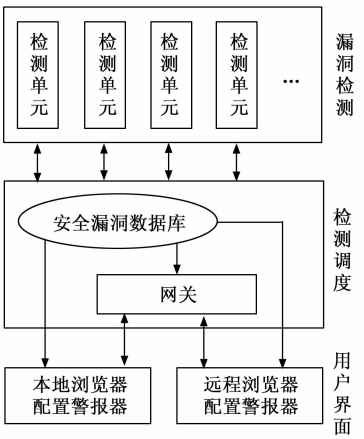

图 2　网络安全漏洞检测器逻辑模型示意图

如图 2 所示，网络安全漏洞检测器逻辑模型主要由 3 个部分构成，分别为用户界面部分、检测调度部分与漏洞检测部分。其中，用户界面部分秉持着简洁易用的原则，为用户提供多种类型的操作方式，满足不同用户的需求。用户可以通过浏览器查询到网络安全漏洞检测结果及其相关信息[7]。与此同时，高级用户还能根据自身需求对漏洞检测程序进行更改与完善，以此来提升网络安全漏洞检测整体性能。

检测调度部分主要是基于安全漏洞数据库与网关传输漏洞检测请求来确定检测目标网络及其其他需求，制定检测单元调度策略，并将其传输给漏洞检测部分，为漏洞检测提供指导作用。

漏洞检测部分主要是通过分布式检测单元对目标网络系统存在的安全漏洞进行检测、识别与预警，与此同时，判定目标网络系统的脆弱性等级，给出相应风险防范措施。根据不同网络用户的需求，编制不同形式的安全漏洞检测报告，并将其反馈给网络安全管理员，其收到反馈结果后，制定相应的安全漏洞补救措施，以保证网络系统的稳定运行，为用户提供更优质的网络环境。

1.3　网络安全漏洞检测器运行模式设计

常规情况下，检测器运行模式主要有两种，分别为单机检测模式与 C/S 模式。当检测器运行模式处于单机检测模式时，只需要管理员对相关参数进行合理配置，即可实现检测器的本地运行，判定网络系统是否存在安全漏洞。需要注意的是，检测器单机检测模式不涉及与服务器的通信过程；当检测器运行模式处于 C/S 模式时，涉及与服务器的通信过程，只有接收到服务器检测指令后才开启漏洞检测单元，漏洞检测结果通过通信方式反馈给网络系统，并将其存储于相应文件中，为后续安全漏洞检测结果查询提供便利[8]。在漏洞检测指令完成后，继续进入监控模式，直到网络用户下线为止。

上述两种运行模式优势与缺陷并存，无法为检测器的稳定运行提供支撑。因此，此研究融合两种运行模式的优势部分，设计新的网络安全漏洞检测器运行模式，具体如图 3 所示。

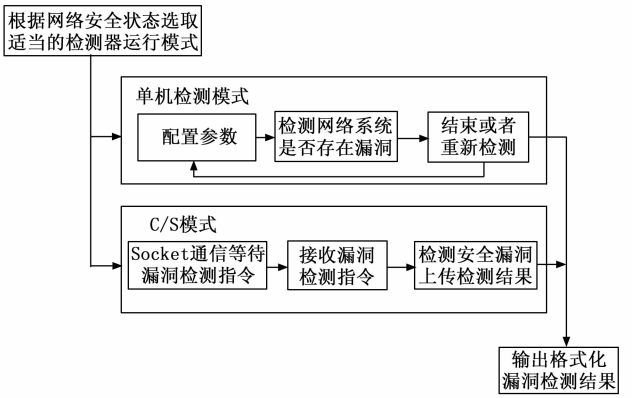

图 3　网络安全漏洞检测器运行模式设计图

如图 3 所示，通过单机检测模式与 C/S 模式的有效结合，可以增加网络安全漏洞检测器运行的安全性以及可靠性[9]。

上述过程从结构、逻辑模型与运行模式 3 个方面出发完成了网络安全漏洞检测器的设计，为最终安全漏洞智能检测的实现提供有力的硬件支撑。

2　网络安全漏洞智能检测系统软件功能实现

2.1　网络安全漏洞知识图谱构建

安全漏洞知识图谱可以表示安全漏洞、网络实体、相关属性等之间的复杂关联关系，对其进行构建可以为后续网络安全漏洞检测提供更多的信息支撑，具体构建过程如下所示。

2.1.1　安全漏洞数据采集与预处理

安全漏洞数据采集与预处理是安全漏洞知识图谱构建的首要环节[10]，也是至关重要的环节。常规情况下，安全漏洞数据主要以非结构化文本形式存在，例如 NVD、CVE 等漏洞数据库，每个漏洞数据库存储的数据种类存在着较大的差异性[11]，使得安全漏洞数据表现形式、存储位置较为随机，为安全漏洞数据采集带来了较大的困难[12]。针对上述安全漏洞数据特点，此研究选取网络爬虫对安全漏洞数据进行采集，具体如图 4 所示。

以图 4 所示程序对网络安全漏洞数据进行采集，并将其整合为集合形式，记为 $X=\{x_1,x_2,\cdots,x_n\}$，其中，n 表示的是网络安全漏洞数据的总数量。网络爬虫在安全漏洞数据采集过程中，容易受到网络环境、恶意程序等干扰，致使安全漏洞数据存在着冗余、层次逻辑混乱、异常等现象，不利于安全漏洞知识图谱的构建，故在安全漏洞知识图谱构建之前，需要对网络安全漏洞数据进行一定的预处理[13]。计算网络安全漏洞数据集合中任意两个数据之间的相似度，表达式为

$$\alpha(x_i,x_j)=\frac{x_i\bigcap x_j}{x_i\bigcup x_j}\times\beta^{\Delta} \tag{1}$$

式中，$\alpha(x_i,x_j)$ 表示的是安全漏洞数据 x_i 与 x_j 之间的相似度；$x_i\bigcap x_j$ 表示的是安全漏洞数据 x_i 与 x_j 的交集；$x_i\bigcup x_j$

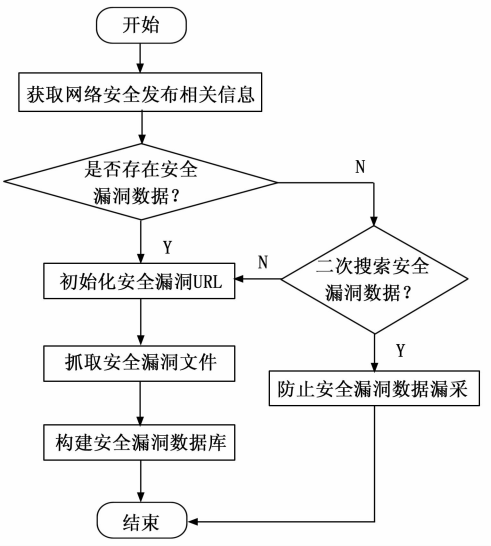

图 4 基于网络爬虫的安全漏洞数据采集程序图

表示的是安全漏洞数据 x_i 与 x_j 的并集；β^3 表示的是安全漏洞数据相似度计算辅助参数，取值范围为 0～1，需要根据网络安全状态实际情况进行科学地设置。

以公式（1）计算结果 $\alpha(x_i,x_j)$ 为基础，判定安全漏洞数据是否为冗余数据，判定规则如下式所示：

$$\begin{cases}\alpha(x_i,x_j)=1 & x_i,x_j \text{ 为冗余数据,删除其中一个}\\ \alpha(x_i,x_j)\neq 1 & x_i,x_j \text{ 为正常数据,保留两个数据}\end{cases} \tag{2}$$

安全漏洞异常数据检测与删除也是其预处理中的关键所在[14]。安全漏洞异常数据检测因子计算公式为：

$$K_e=\frac{x_i-\bar{x}}{\sigma_x} \tag{3}$$

式中，K_e 表示的是安全漏洞异常数据检测因子；$\bar{x}$ 表示的是安全漏洞数据的平均值；σ_x 表示的是安全漏洞数据的标准差数值。

以公式（3）计算结果 K_e 为基础，判定安全漏洞数据是否为异常数据，判定规则如下式所示：

$$\begin{cases}|K_e|>1 & x_i \text{ 为异常数据}\\ |K_e|\leqslant 1 & x_i \text{ 为正常数据}\end{cases} \tag{4}$$

式中，$|K_e|$ 表示的是安全漏洞异常数据检测因子 K_e 的绝对值。

将检测到的冗余数据与异常数据进行删除处理，并将剩余数据进行重新整合，即可获得预处理后的网络安全漏洞数据集合，记为 $X'=\{x'_1,x'_2,\cdots,x'_m\}$，其中，$m$ 表示的是预处理后网络安全漏洞数据的总数量[15]。

上述过程完成了安全漏洞数据的采集与预处理，并获得了最终的网络安全漏洞数据集合 $X'=\{x'_1,x'_2,\cdots,x'_m\}$，为后续安全漏洞知识图谱实体的识别奠定坚实的基础。

2.1.2 安全漏洞知识图谱实体识别

安全漏洞知识图谱实体识别主要是针对安全漏洞实体进行识别，是知识图谱构建的主要依据之一。在网络运行实际情况下，每个安全漏洞均具备独一无二的 ID，其对应的属性信息也存在着较大的差异性。由此可见，能够根据属性信息对安全漏洞知识图谱实体进行精准识别。

常规情况下，安全漏洞属性信息主要包括漏洞风险数值、漏洞文件名称、漏洞编程语言、漏洞爆发点等，为了方便后续漏洞实体的识别，对属性信息进行统计，具体如表 1 所示。

表 1 安全漏洞属性信息统计表

属性名称	属性排序	属性描述
CVE-ID	3	安全漏洞 ID
CVSS	1	安全漏洞危险等级
PathName	5	安全漏洞文件名称
Language	7	安全漏洞源代码编程语言
Breakpoint	2	安全漏洞爆发点
Grain	8	安全漏洞粒度
Entry	10	安全漏洞入口
PublishDate	4	安全漏洞发布时间
Exit	9	安全漏洞出口
Description	6	安全漏洞相关描述

如表 1 内容所示，每个安全漏洞均是由多个属性信息构成的，以此为基础，衡量未知安全漏洞与已知安全漏洞属性信息之间的相关系数[16]，计算公式为：

$$\chi(P_i,Q_j)=\frac{\alpha(P_i,Q_j)}{N}+\varepsilon_e \tag{5}$$

式中，$\chi(P_i,Q_j)$ 表示的是未知安全漏洞属性信息集合 P_i 与已知安全漏洞属性信息集合 Q_j 之间的相关系数；$\alpha(P_i,Q_j)$ 表示的是未知安全漏洞与已知安全漏洞属性信息的相似度；N 表示的是属性信息的总数量；ε_e 表示的是误差调整项，承担着提升相关系数精度的任务。

以公式（5）计算结果 $\chi(P_i,Q_j)$ 为基础，制定安全漏洞知识图谱实体识别规则，具体如下式所示：

$$\begin{cases}\chi(P_i,Q_j)\geqslant\hat{\delta} & P_i \text{ 是知识图谱实体}\\ \chi(P_i,Q_j)<\hat{\delta} & P_i \text{ 非知识图谱实体}\end{cases} \tag{6}$$

式中，$\hat{\delta}$ 表示的是安全漏洞知识图谱实体识别阈值，其需要根据安全漏洞属性信息实际情况来设置。

通过上述过程完成了安全漏洞知识图谱实体的精准识别，为最终知识图谱的构建做好充足的准备工作。

2.1.3 安全漏洞知识图谱关系抽取

在网络实际运行过程中，安全漏洞知识图谱主要存在 4 种依赖关系，分别为函数调用依赖关系、控制依赖关系、声明依赖关系与数据流依赖关系，其是知识图谱构建的基础与前提之一[17]。因此，此节对上述关系进行描述与抽取。

为了方便后续安全漏洞知识图谱关系的描述以及抽取，设置安全漏洞知识图谱任意两个节点为 R_1 与 R_2，具体关系抽取过程如下所示：

1）函数调用依赖关系抽取：

当安全漏洞知识图谱节点 R_1 被调用至节点 R_2 边缘时，认定两者之间关系为函数调用依赖关系，将其记为 R_1

$\xrightarrow{PCD} R_2$；

2）控制依赖关系抽取：

当安全漏洞知识图谱节点 R_2 隶属于节点 R_1，并且需要在节点 R_1 的帮助下才能进行相关操作时，认定两者之间关系为控制依赖关系，记为 $R_1 \xrightarrow{CD} R_2$；

3）声明依赖关系抽取：

当声明某变量过程中均涉及了安全漏洞知识图谱节点 R_1 与 R_2，则表明两者之间关系为声明依赖关系，其是一种特殊关系，记为 $R_1 \xrightarrow{SD} R_2$；

4）数据流依赖关系抽取：

当安全漏洞知识图谱节点 R_1 与 R_2 之间存在特定路径，并且节点 R_1 与 R_2 变量定义一致，则表明两者之间关系为数据流依赖关系，记为 $R_1 \xrightarrow{DFD} R_2$。

依据上述描述在安全漏洞知识图谱节点中进行搜索、识别与抽取，为后续安全漏洞知识图谱可视化处理提供支撑。

2.1.4　安全漏洞知识图谱可视化

以上述安全漏洞知识图谱实体识别结果与关系抽取结果为依据，定义安全漏洞知识图谱表示形式，设计安全漏洞知识图谱结构，从而实现安全漏洞知识图谱的构建与可视化。

此研究采用三元组表示安全漏洞知识图谱，表达式为

$$G = (X', P, R) \tag{7}$$

式中，G 表示的是安全漏洞知识图谱三元组表示形式；X' 表示的是安全漏洞数据集合；P 表示的是安全漏洞知识图谱实体集合；R 表示的是安全漏洞知识图谱关系集合。

安全漏洞知识图谱主要包含两大结构，分别为漏洞实体结构与其他实体结构[18]。其中，漏洞实体结构中包含着安全漏洞属性信息、基本信息等，其他实体结构中包含着网络安防实体、网络运行程序实体等。安全漏洞知识图谱构建结果如图 5 所示。

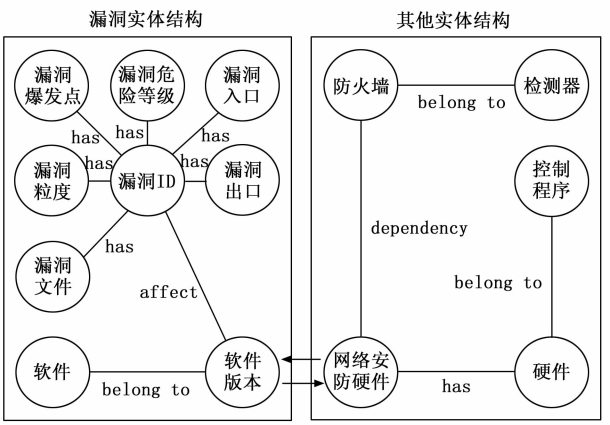

图 5　安全漏洞知识图谱构建结果示意图

上述过程完成了网络安全漏洞知识图谱的构建与可视化，为网络安全漏洞智能检测系统的实现打下坚实的基础。

2.2　网络安全漏洞智能检测功能实现

以上述网络安全漏洞知识图谱构建结果与网络安全漏洞检测器设计结果为依据，构建网络安全漏洞智能检测整体架构，制定网络安全漏洞智能检测具体流程（检测器软件程序），从而获取最终网络安全漏洞智能检测结果，为网络系统的稳定运行提供保障。

网络安全漏洞智能检测整体架构如图 6 所示。

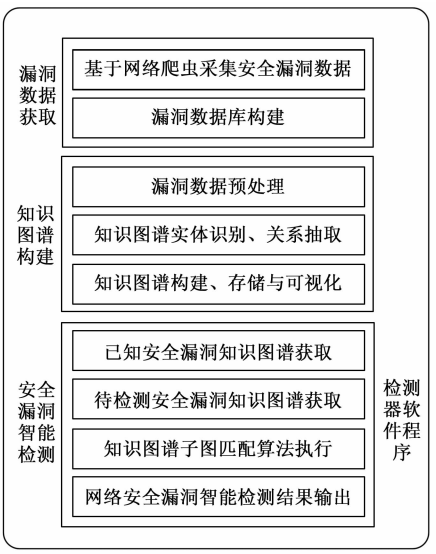

图 6　网络安全漏洞智能检测整体架构图

如图 6 所示，网络安全漏洞智能检测主要依据知识图谱子图匹配算法实现，简单地说，就是在已知安全漏洞知识图谱 $F = (\widetilde{X}', \widetilde{P}, \widetilde{R})$ 匹配同构于待检测安全漏洞知识图谱 $G = (X', P, R)$ 的全部数据子图[19]。从本质角度出发，知识图谱是一个有向多标签图，顶点之间存在着多条边。与普通图谱相比较，知识图谱内部结构更加稠密，包含更多的关系信息。为了提升网络安全漏洞智能检测的时间效率，提出新的知识图谱子图匹配算法——FGqT-Match 算法，其主要划分为两个阶段，阶段一为 FGqT 索引构造，阶段二为最优匹配序列获取。

基于 FGqT-Match 算法的网络安全漏洞智能检测程序如下所示。

阶段一：FGqT 索引构造。

FGqT 索引（流图索引）主要是基于漏洞数据匹配顶点对构造而成。其中，顶点对主要描述的是待检测安全漏洞知识图谱顶点 u 到已知安全漏洞知识图谱顶点 v 的映射函数，记为 $[u, v]$。需要注意的是，在漏洞数据匹配顶点对应用之前，需对其是否匹配进行精准验证，验证规则如下式所示：

$$\begin{cases} L_1(u) = L_1(v) \\ L_2(u, u') = L_2(v, v') \\ \forall u' \in G = (X', P, R), \forall v' \in F = (\widetilde{X}', \widetilde{P}, \widetilde{R}) \end{cases} \tag{8}$$

式中，$L_1(u)$ 与 $L_1(v)$ 表示的是漏洞数据匹配顶点 u 与 v 对应的标签函数；$L_2(u, u')$ 与 $L_2(v, v')$ 表示的是漏洞数据匹配边 (u, u') 与 (v, v') 对应的标签函数；u' 表示的是待检测

安全漏洞知识图谱中与顶点 u 连接的任意一个顶点；v' 表示的是已知安全漏洞知识图谱中与顶点 v 连接的任意一个顶点。

当漏洞数据匹配顶点对 $[u,v]$ 满足公式（8）全部约束条件时，则表示漏洞数据顶点 u 与 v 是匹配的，对其进行保留处理；当漏洞数据匹配顶点对 $[u,v]$ 不符合公式（8）某一条约束条件时，则表示漏洞数据顶点 u 与 v 是不匹配的，对其进行删除处理。

一个 FGqT 索引是由多个顶点对构成的（顶点对是匹配的），将其记为 $M=\{[u_1,v_1],[u_2,v_2],\cdots,[u_n,v_n]\}$，其中，$n$ 表示的是漏洞数据匹配顶点对的总数量。

阶段二：最优匹配序列获取。

以上述构造的 FGqT 索引 $M=\{[u_1,v_1],[u_2,v_2],\cdots,[u_n,v_n]\}$ 作为依据，计算待检测安全漏洞知识图谱顶点中标签数量与已知安全漏洞知识图谱顶点中标签数量比值，将其记为匹配概率，表达式为

$$\zeta=\frac{|G|\cdot f(u)}{|F|\cdot f(v)} \tag{9}$$

式中，ζ 表示的是匹配概率；$|G|$ 与 $|F|$ 表示的是待检测安全漏洞知识图谱与已知安全漏洞知识图谱中顶点候选集的大小；$f(u)$ 与 $f(v)$ 表示的是待检测安全漏洞知识图谱与已知安全漏洞知识图谱顶点中标签数量。

以公式（9）计算结果 ζ 为基础，判定当前子图匹配序列 $\{[u_1,v_1],[u_2,v_2],\cdots,[u_n,v_n]\}$ 是否为最优匹配序列，具体判定规则如下式所示：

$$\begin{cases}\zeta\geqslant\Psi & \text{是}\\ \zeta<\Psi & \text{否}\end{cases} \tag{10}$$

式中，Ψ 表示的是最优子图匹配序列判定阈值，其需要根据安全漏洞知识图谱实际情况进行具体的设置。

当不存在最优子图匹配序列时，表明网络系统中未存在安全漏洞；当存在最优子图匹配序列时，表明网络系统中存在安全漏洞，并且安全漏洞类型为子图匹配到的已知安全漏洞知识图谱对应的安全漏洞[20]。

综上所述，在知识图谱原理的应用下，实现了网络安全漏洞智能检测系统的设计与运行，最终获取网络安全漏洞检测结果。

3 实验与结果分析

设置文献［1］提出的黑盒遗传算法、文献［2］提出的 N-gram 算法、文献［3］提出的被动分簇算法作为对比系统 1、对比系统 2 与对比系统 3，联合设计系统共同进行网络安全漏洞智能检测对比实验，以此来验证设计系统的应用效果。

3.1 实验工况设置

选取某局域网络系统作为实验对象，其覆盖范围较小，方便实验的进行。单一实验工况背景下，无论怎么增加实验次数，获得的实验结论可信度均较低。因此，为了提升此研究实验结论的可信度，在实验进行之前，依据实验对象——局域网络系统实际情况，设置 10 种实验工况，具体如表 2 所示。

表 2 实验工况设置表

实验工况	是否存在安全漏洞	安全漏洞数量/个	漏洞来源
1	是	15	管理脆弱性、技术脆弱性
2	是	23	管理脆弱性、系统脆弱性
3	否	0	—
4	是	10	技术脆弱性
5	否	0	—
6	是	25	管理脆弱性、技术脆弱性
7	是	26	管理脆弱性
8	是	30	管理脆弱性、技术脆弱性与系统脆弱性
9	否	0	—
10	是	14	管理脆弱性、系统脆弱性

如表 2 内容所示，设置的 10 种实验工况中，安全漏洞数量、漏洞来源等均不一致，表明每种实验工况对应的背景环境存在着较大的差异性，符合设计系统应用效果的测试需求。

3.2 评价指标选取

为了直观显示设计系统的应用效果，选取适当的评价指标。从安全漏洞检测精度与效率两个角度出发，选取网络安全漏洞漏检率、网络安全漏洞检测 F1 值与网络安全漏洞检测响应时间作为评价指标。其中，前两个评价指标表征的是安全漏洞检测精度，后一个评价指标表征的是安全漏洞检测效率。

网络安全漏洞漏检率计算公式为

$$O=\frac{h_{\text{total}}-h_1}{h_{\text{total}}}\times 100\% \tag{11}$$

式中，O 表示的是网络安全漏洞漏检率，其与安全漏洞检测性能呈现显著的反比例关系，即漏检率数值越小，表明安全漏洞检测性能越好；h_{total} 表示的是网络系统中存在的安全漏洞总数量；h_1 表示的是检测到的安全漏洞数量。

网络安全漏洞检测 F1 值计算公式为

$$F1=2\times\frac{p\times r}{p+r} \tag{12}$$

式中，$F1$ 表示的是融合了精确率与召回率的评价因子，能更准确地衡量网络安全漏洞检测质量，该数值越大表明安全漏洞检测性能越好；p 表示的是安全漏洞检测精确率；r 表示的是安全漏洞检测召回率。

网络安全漏洞检测响应时间计算公式为

$$T=T_1-T_0 \tag{13}$$

式中，T 表示的是网络安全漏洞检测响应时间，该数值越小表明安全漏洞检测性能越好；T_1 表示的是网络安全漏洞检测结果输出时间；T_0 表示的是网络安全漏洞出现时间。

上述过程完成了评价指标的选取与计算公式介绍，为后续实验结果分析提供依据。

3.3 实验结果分析

3.3.1 网络安全漏洞检测精度分析

在实验工况设置及评价指标选取的基础上，进行网络

安全漏洞检测实验。根据表 2 中已知的安全漏洞信息，确定安全漏洞知识图谱的表示形式。设计安全漏洞知识图谱的结构。将安全漏洞的关键属性和关系以节点和边的形式进行建模，并确定节点之间的关联关系。利用已构建的安全漏洞知识图谱，辅助识别网络中存在的潜在漏洞问题。采集检测过程中网络系统中存在的安全漏洞总数量及检测到的安全漏洞数量，代入公式（11）获得网络安全漏洞漏检率实验结果，如表 3 所示。根据公式（11）计算安全漏洞检测精确率及安全漏洞检测召回率，代入公式（12）获得网络安全漏洞检测 F_1 值实验结果，如表 4 所示。

通过实验获得网络安全漏洞检测精度评价指标——网络安全漏洞漏检率、网络安全漏洞检测 F_1 值如表 3、表 4 所示。

表 3　网络安全漏洞漏检率　　%

实验工况	设计系统	对比系统 1	对比系统 2	对比系统 3
1	3.12	10.60	11.45	15.89
2	2.00	12.45	12.56	15.45
3	1.23	11.03	10.08	14.78
4	3.56	9.89	12.78	16.25
5	4.15	12.45	10.88	13.56
6	3.26	16.89	13.89	12.02
7	4.59	12.00	9.78	10.09
8	5.12	15.78	8.45	11.45
9	2.01	13.25	15.40	12.45
10	1.56	16.45	12.02	9.56

如表 3 数据所示，在不同实验工况背景条件下，应用设计系统获得的网络安全漏洞漏检率低于对比系统 1、对比系统 2 与对比系统 3，其最小值达到了 1.23%，而对比系统 1、对比系统 2 与对比系统 3 的网络安全漏洞漏检率最小值分别为 9.89%、8.45%、9.56。设计系统的网络安全漏洞漏检率较低，是因为设计系统通过对真实数据进行知识图谱的建模和构建，通过定义安全漏洞知识图谱的表示形式和结构，并进行可视化展示，系统可以更清晰地呈现安全漏洞之间的关联信息。系统利用属性信息识别安全漏洞实体，移除冗余数据和异常数据，这有助于用户理解和分析漏洞之间的联系，提高漏洞的发现和推断能力，从而提高了检测的准确性和可靠性。

表 4　网络安全漏洞检测 F_1 值

实验工况	设计系统	对比系统 1	对比系统 2	对比系统 3
1	8.56	5.23	6.23	7.12
2	9.12	4.10	4.12	4.02
3	9.15	5.23	5.16	2.01
4	9.50	5.10	5.49	1.56
5	8.45	5.59	5.78	3.46
6	7.10	5.45	4.02	4.15
7	8.52	4.12	4.30	4.89
8	8.49	4.89	4.90	4.75
9	8.79	4.78	5.15	5.42
10	9.15	3.09	5.23	5.25

如表 4 数据所示，在不同实验工况背景条件下，应用设计系统获得的网络安全漏洞检测 F_1 值高于对比系统 1、对比系统 2 与对比系统 3，其最大值达到了 9.50，而对比系统 1、对比系统 2 与对比系统 3 的网络安全漏洞检测 F_1 值最大值分别为 5.59、6.23、7.12。设计系统的网络安全漏洞检测 F_1 值较高，意味着漏洞检测系统在同时考虑了准确率和召回率的情况下取得了更好的平衡，F_1 值高说明系统可以精确地识别安全漏洞，较少将正常的网络流量误判为漏洞或错误地报告漏洞。这有助于避免误报、减少虚假警报，让系统运行更加可靠和稳定。F_1 值高也意味着系统能够高效地检测出网络中的真实漏洞，尽可能地避免漏报。提高召回率意味着漏洞检测系统能够更多地识别到实际存在的漏洞，及时采取措施进行修复和防护。

3.3.2　网络安全漏洞检测效率分析

根据上述实验过程，记录网络安全漏洞检测结果输出时间及网络安全漏洞出现时间，代入公式（13）获得网络安全漏洞检测效率实验结果，如图 7 所示。

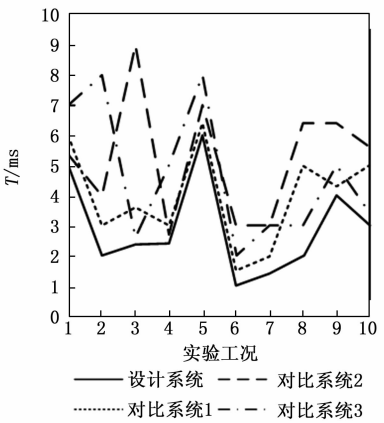

图 7　网络安全漏洞检测响应时间示意图

如图 7 数据所示，在不同实验工况背景条件下，应用设计系统获得的网络安全漏洞检测响应时间均低于对比系统 1、对比系统 2 与对比系统 3，其最小值达到了 1 ms，而对比系统 1、对比系统 2 与对比系统 3 的检测响应时间最小值分别为 1.5 ms、2 ms、2.8 ms。表明设计系统的网络安全漏洞检测效率更高。设计系统通过对大量真实数据进行预处理，去除冗余数据和异常数据，提高数据的质量和准确性。这样可以使得系统对漏洞进行快速检测和分析，从而减少响应时间。通过整体架构的设计和具体流程的制定，使得漏洞检测能够高效运行。

4　结束语

随着网络技术的发展与普及，网络已经成为人们生产与生活中必不可缺的事物。随之而来的网络安全问题也日益严重，威胁着用户信息的安全与网络系统的稳定，成为制约网络后续发展的关键所在，故提出基于知识图谱的网络安全漏洞智能检测系统设计研究。通过实验得出，所设计系统降低了网络安全漏洞漏检率，提升了网络安全漏洞

检测 F_1 值，缩短了网络安全漏洞检测响应时间，为安全漏洞检测提供更有效的系统支撑。综上所述，基于知识图谱的网络安全漏洞智能检测系统可以为网络安全领域带来更高效、精准和智能的漏洞检测和防护能力，也能够为相关研究提供一定的借鉴作用。

参考文献：

[1] 王小虎，王 超，李 群，等. 基于黑盒遗传算法的电力系统网络安全漏洞挖掘方法 [J]. 沈阳工业大学学报，2021，43 (5)：500－504.

[2] 蒋荣萍. 基于 N-gram 算法的网络安全风险检测系统设计 [J]. 现代电子技术，2021，44 (1)：25－28.

[3] 张 杰，景 雯，陈 富. 基于被动分簇算法的即时通信网络协议漏洞检测 [J]. 吉林大学学报 (工学版)，2021，51 (6)：2253－2258.

[4] 熊 强，杨欣琦，李治文. 网络安全漏洞信息披露中多元参与主体行为策略演化博弈分析 [J]. 运筹与管理，2021，30 (7)：102－109.

[5] 李明磊，黄 晖，陆余良，等. SymFuzz：一种复杂路径条件下的漏洞检测技术 [J]. 计算机科学，2021，48 (5)：25－31.

[6] 吴昊天. 浅谈煤炭企业网络安全防护 [J]. 工矿自动化，2021，47 (S2)：165－167.

[7] 倪雄军，李健俊，李钰靓，等. 卷接设备 IPC 控制系统网络安全监测模型的构建 [J]. 烟草科技，2022，55 (1)：99－106.

[8] 吴海涛，代尚林，乔中伟，等. 基于 RBF－SVM 智能配变终端的网络安全态势评估 [J]. 电力科学与技术学报，2021，36 (5)：35－40.

[9] 陈伟雄，杨晓晨，春增军，等. 电力企业网络安全威胁情报管理体系的研究与实践 [J]. 电信科学，2022，38 (7)：184－189.

[10] 文 敏，王荣存，姜淑娟. 基于关系图卷积网络的源代码漏洞检测 [J]. 计算机应用，2022，42 (6)：1814－1821.

[11] 程靖云，王布宏，罗 鹏. 基于图表示和 MHGAT 的代码漏洞静态检测方法 [J]. 系统工程与电子技术，2023，45 (5)：1535－1543.

[12] 何 杰，蔡瑞杰，尹小康，等. 面向 Cisco IOS-XE 的 Web 命令注入漏洞检测 [J]. 计算机科学，2023，50 (4)：343－350.

[13] 王 剑，匡洪宇，李瑞林，等. 基于 CNN-GAP 可解释性模型的软件源码漏洞检测方法 [J]. 电子与信息学报，2022，44 (7)：2568－2575.

[14] 张 铮，张星娜，吕 卓，等. 基于深度学习的智能合约漏洞检测方法 [J]. 重庆邮电大学学报：自然科学版，2022，34 (5)：914－920.

[15] 印桂生，高 乐，庄 园，等. 基于字节码关键路径的智能合约漏洞检测 [J]. 哈尔滨工程大学学报，2022，43 (2)：255－261.

[16] 陈开阳，徐 凡，王明文. 基于知识图谱和图像描述的虚假新闻检测研究 [J]. 江西师范大学学报 (自然科学版)，2021，45 (4)：398－402.

[17] 史运涛，刘 召，李书钦，等. 基于知识图谱注意力网络的食品安全风险评估模型 [J]. 食品工业，2021，42 (12)：471－475.

[18] 郭军军，王 乐，王正源，等. 软件安全漏洞知识图谱构建方法 [J]. 计算机工程与设计，2022，43 (8)：2137－2145.

[19] 王海晏，江 涛，王 芳，等. 基于知识图谱的目标识别模型 [J]. 探测与控制学报，2022，44 (6)：76－80.

[20] 丁兆云，刘 凯，刘 斌，等. 网络安全知识图谱研究综述 [J]. 华中科技大学学报：自然科学版，2021，49 (7)：79－91.

(上接第 62 页)

[7] 胡哲纲，车伟扬. 分布式 UPS 蓄电池远程在线监测系统 [J]. 电子世界，2020，597 (15)：164－165.

[8] 夏志梁. UPS 设备在线监测技术及应用 [J]. 铁道通信信号，2018，54 (10)：33－35.

[9] 钱承山，宗文杰，孙 宁，等. 基于 NB-IoT 的智慧消防栓监测系统设计 [J]. 国外电子测量技术，2021，40 (12)：151－158.

[10] 胡各优，张钦科，胡 辉，等. 基于 SUI-101A 的用电器分析识别装置设计 [J]. 现代信息科技，2022，6 (9)：66－68.

[11] 窦宏博，吴建峰，徐 静，等. 变电站直流系统铅酸电池内阻在线测量方法 [J]. 电池，2020，50 (2)：165－167.

[12] 邓已媛，陈 松，范仰栋，等. 基于 INA282 的集成电压电流检测电路设计与优化 [J]. 电子器件，2019，42 (4)：953－957.

[13] 付文新，王洪丰. 基于 STM32 单片机和 DHT11 温湿度传感器的温湿度采集系统的设计与实现 [J]. 光源与照明，2022 (3)：119－121.

[14] LIANG C T，LIANG L P，WANG Z J. A fully integrated digital LDO with voltage peak detecting and push－pull feedback loop control [J]. IEICE Electronics Express，2018，15 (15)：20180611.

[15] 陈 阳，崔仁胜，朱小毅，等. 物联网 MQTT 协议在地震波形实时传输中的应用 [J]. 地球物理学进展，2020，35 (4)：1232－1237.

[16] 朱代先，王力立，刘冰冰，等. 基于 NB-IoT 的智慧井盖监测系统设计与实现 [J]. 计算机测量与控制，2019 (10)：55－59.

[17] 赵 靖. 基于 STM32 的具有谐波分析功能的智能电表设计 [D]. 上海：上海交通大学，2012.

[18] 黄冬梅，唐志涛，张晋轩，等. 基于 FFT 的电力谐波分析方法 [J]. 电工技术，2022 (20)：213－216.

[19] 杨柳林，谢振林. 基于窄带物联网的电缆接头温度监测系统 [J]. 科学技术与工程，2022，22 (6)：2275－2283.

[20] 张 文，黎昌金. 电视发射机房 UPS 蓄电池远程在线监测管理系统 [J]. 计算机测量与控制，2012，20 (11)：2929－2931.

[21] 姚金坤. UPS 及直流电源设备在线监测维护管理系统 [J]. 石油化工自动化，2005 (1)：16－18.