一种新的基于形式概念分析的漏洞分析预处理方法

王绍杰，祁　斌，万佳蓉
（中国电子信息产业集团有限公司第六研究所，北京　100083）

摘　要：漏洞分析技术已经成为企业应对安全问题的一个研究重点，通过对系统中已有漏洞进行分析，可以帮助企业和相关工作人员熟悉漏洞的特点与产生原因，甚至可以快速、高效地发现未知漏洞，避免资产损失。针对漏洞分析技术中漏洞预处理过程展开研究，提出一种新的基于形式概念分析的漏洞分析预处理方法，实现了对已有漏洞根据其特征快速进行聚类，并以图形化界面直观展示给用户的功能。实验表明新的漏洞分析预处理方法可以快速、准确地获得漏洞之间的所有关联，为用户进一步分析漏洞提供基础，符合企业及相关人员使用需求。

关键词：漏洞分析；数据预处理；形式概念分析
中图分类号：TP393. 08　　**文献标识码**：A　　**DOI**：10. 19358/j. issn. 2097 - 1788. 2023. 08. 006
引用格式：王绍杰，祁斌，万佳蓉. 一种新的基于形式概念分析的漏洞分析预处理方法［J］. 网络安全与数据治理，2023，42(8)：34 - 39.

A new preprocessing method for vulnerability analysis based on formal concept analysis

Wang Shaojie，Qi Bin，Wan Jiarong
(The 6th Research Institute of China Electronics Corporation，Beijing 100083，China)

Abstract：Vulnerability analysis technology has become a research focus for enterprises to deal with security issues. By analyzing the existing vulnerabilities in the system，it can help enterprises and related staff to become familiar with the characteristics and causes of vulnerabilities，and even discover unknown vulnerabilities quickly and efficiently to avoid asset loss. This paper conducts research on the vulnerability preprocessing process in vulnerability analysis technology，and proposes a new vulnerability analysis preprocessing method based on formal concept analysis，which realizes the function of rapidly clustering existing vulnerabilities according to their vulnerability characteristics，and visually displaying them to users with graphical interface. Experimental results show that the new vulnerability analysis preprocessing method can quickly and accurately obtain all the correlations between vulnerabilities，which can provide a basis for users to further analyze vulnerabilities，and meet the needs of enterprises and related personnel.

Key words：vulnerability analysis；data preprocessing；formal concept analysis

0　引言

互联网为工业领域提供了数据化管理、绿色生产等高效、精准、智能化的生产方式，使得企业可以更好地适应市场变化和客户需求，提高竞争力和生产效率。但也存在一些潜在的弊端，工业系统中计算机网络变得越来越复杂，利用系统中漏洞进行网络攻击的现象时有发生，工业控制系统成为工业网络攻击的首要目标，因此产业界对工业控制安全漏洞的发现与管理工作异常重视[1]。

漏洞所产生的严重后果使企业和相关人员将更多精力放在漏洞相关技术的研究上[2-3]，漏洞分析技术也得到了广泛关注[4-7]。通过深入分析已经发现的漏洞，对挖掘、修补未知漏洞具有重要意义。因此，已知的漏洞信息也得到了相关企业的关注，中国信息安全测评中心于2009年创建了国家信息安全漏洞库CNNVD，旨在通过该库收集、分析和评估计算机系统中的安全漏洞，并向公众发布警报。在2010年，国家计算机网络应急技术处理协调中心建立了CNVD网站，使得漏洞信息可以更广泛地传播和共享[8]。随后，在2020年国家为规范网络安全漏洞信息发布了一系列标准[9-11]，如何正确利用已有的安全漏洞信息对企业发现和处理未知漏洞具有重要意义。

形式概念分析（Formal Concept Analysis，FCA）是德

国数学家 Wille 于上世纪八十年代为重构概念格提出的理论[12]。该理论自提出至今，得到了很大发展，已经被广泛应用在数据分析[13]、软件工程[14]、信息检索[15]和推荐算法[16]等领域。

在漏洞分析技术中使用形式概念分析方法完成信息预处理，有利于将漏洞进行精准聚类，探索不同漏洞间的共同特征，帮助漏洞研究人员更深入地了解漏洞的本质和特点，从而更快地发现系统漏洞，建立起针对不同漏洞类型的检测、防御和修复方法。

1 形式概念分析简述

形式概念分析是一种用于分析和组织复杂数据集的数学框架，是分析对象及其属性之间逻辑关系的一种方式。概念的内涵和外延是概念具有的两个基本特征，概念的内涵指这个概念的含义，即该概念所反映的事物对象所持有的属性；概念的外延是指这个概念所反映的事物对象的范围，即具有概念所反映的属性的事物或对象[17]。下面先给出形式概念分析的相关定义。

定义 1[17] 设 (G, M, I) 为一个形式背景，其中 $G=\{g_1, \cdots, g_p\}$ 为对象集，每个 g_i $(i\leqslant p)$ 称为一个对象；$M=\{m_1, \cdots, m_q\}$ 为属性集，每个 m_j $(j\leqslant q)$ 称为一个属性；I 为 G 与 M 之间的二元关系，$I\subseteq G\times M$。若 $(g, m)\in I$，则表示对象 g 具有属性 m，也记为 gIm。

形式背景可以通过二维数组表示，本文将对象 g 具有属性 m 记为 1，将对象 g 不具有属性 m 记为 0。

定义 2[17] 对于形式背景 (G, M, I)，对任意的 $X\subseteq G$，$B\subseteq M$，都可以定义一对对偶算子：

$$X^*=\{m \mid m\in M, \forall g\in X, \text{gIm}\}$$

$$B^*=\{g \mid g\in G, \forall m\in B, \text{gIm}\}$$

其中，X^* 表示 X 中所有对象共同具有的属性集合，B^* 表示共同具有 B 中所有属性的对象集合。

例 1 表 1 是一个形式背景 (G, M, I)，其中对象集 $G=\{1, 2, 3, 4, 5, 6\}$，属性集 $M=\{a, b, c, d, e, f\}$。

表 1 形式背景 (G, M, I)

	a	b	c	d	e	f
1	1	0	1	0	1	0
2	0	1	0	0	0	0
3	0	0	0	1	0	0
4	1	0	1	0	0	0
5	1	1	1	1	0	0
6	0	1	0	0	0	1

性质 1 设 (G, M, I) 为一个形式背景，对任意的 X_1，X_2，均有 $X\subseteq G$；B_1，B_2 均有 $B\subseteq M$，可以得到以下基本结论：

(1) $X_1\subseteq X_2\rightarrow X_2^*\subseteq X_1^*$，$B_1\subseteq B_2\rightarrow B_2^*\subseteq B_1^*$；

(2) $X\subseteq X^{**}$，$B\subseteq B^{**}$；

(3) (X^{**}, X^*) 和 (B^*, B^{**}) 都是概念。

根据表 1 中所示的形式背景 (G, M, I)，可以计算得到的概念为 $C_1=(\{1, 2, 3, 4, 5, 6\}, \{\})$，$C_2=(\{1, 4, 5\}, \{a, c\})$，$C_3=(\{\}, \{a, b, c, d, e, f\})$，$C_4=(\{5\}, \{a, b, c, d\})$，$C_5=(\{1\}, \{a, c, e\})$，$C_6=(\{2, 5, 6\}, \{b\})$，$C_7=(\{3, 5\}, \{d\})$，$C_8=(\{6\}, \{b, f\})$。

形式概念分析以概念格为核心数据结构，通过使用概念格可以清晰表示出形式背景中具有相同对象的属性集和具有相同属性对象集之间的关系。例 1 中形式背景生成的概念格如图 1 所示。

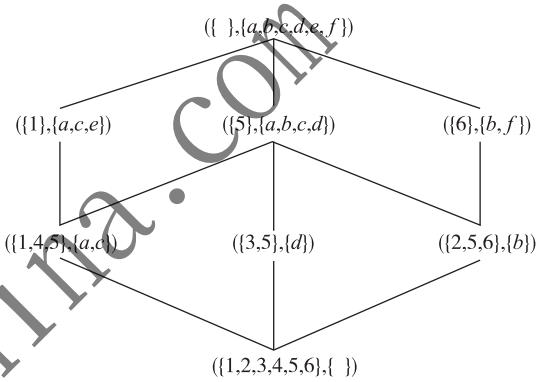

图 1 例 1 形式背景的概念格

2 基于形式概念分析的漏洞分析预处理技术

形式概念分析是一种基于格论的知识表示和分析方法，可以帮助发现对象之间的关系、属性之间的关联以及潜在的模式和规律。通过将漏洞信息转化为形式背景，并利用形式概念分析方法对漏洞特征进行关联分析，可以发现不同漏洞之间的潜在联系，以便相关工作人员根据具有关联的漏洞进行深入分析，进一步挖掘不同漏洞之间的共有特征，理解漏洞的本质和相关原理，从而为预防或发现漏洞提供帮助。下面将介绍如何对漏洞信息进行分析预处理。

2.1 基于漏洞特征的形式背景设计

如果将漏洞的特征信息都存储到形式背景 (G, M, I) 中，其中 $G=\{g_1, g_2, \cdots, g_p\}$，$M=\{m_1, m_2, \cdots, m_q\}$，那么对于任意被管理的漏洞都能在形式背景中找到表示它的对象（即漏洞名称）g_i $(i\leqslant p)$，对于任意用于聚类的漏洞特征都能在形式背景中找到表示它的属性（即特征描述）m_j $(j\leqslant q)$，gIm 则表示漏洞 g 具有漏洞特征 m。

为了方便数据处理，将漏洞具有的特征信息均转为漏洞与特征之间的二元关系，并以二值化描述，值为 1 时表示漏洞具有该漏洞特征，为 0 时则表示不具有该漏洞特征。根据《信息安全技术 网络安全漏洞分类分级指

南》（GB/T 30279—2020）[9] 中的漏洞评级标准，本文从10 个特征维度对漏洞进行分析，每个特征维度所包含的评定等级如表 2 所示。为方便表述，在下文中均以特征维度序号表示漏洞特征。

表 2　漏洞特征等级表

序号	漏洞特征	评定等级	序号	漏洞特征	评定等级
1	访问路径	网络、邻接、本地、物理	6	完整性	严重、一般、无
2	触发要求	低、高	7	可用性	严重、一般、无
3	权限需求	无、低、高	8	被利用成本	低、中、高
4	交互条件	不需要、需要	9	修复难度	高、中、低
5	保密性	严重、一般、无	10	影响范围	高、中、低、无

根据表 2 中漏洞特征可以轻松将任意漏洞信息转为形式背景。例如已知某漏洞 X 可以通过网络远程触发漏洞但触发要求高，需要一般权限，不需要人机交互即可使用该漏洞，但是该漏洞对受影响实体承载信息的保密性影响严重，不影响承载信息的完整性和可用性。在当前全球互联网环境下，漏洞开发成本低，易于修复，影响范围广。可以将上述信息转为仅有一个对象的形式背景，如表 3 所示。

表 3　漏洞 X 的形式背景

	1	2	3	4	5	6	7	8	9	10
X	1 000	01	010	10	100	001	001	100	001	1 000

由于形式背景是二值背景，而每个特征维度中包含多个评定等级，因此将每个评定等级对应 1 个属性，用 1 个位数值表示。例如表 3 中，特征维度为 1 的值为 1 000。特征维度为 1 表示访问路径，共 4 个位数值，第 1 位表示访问路径为网络，第 2 位表示访问路径为邻接，第 3 位表示访问路径为本地，第 4 位表示访问路径为物理。而值 1 000 中仅第 1 位为 1，表示访问路径为网络。特征维度 2 的值为 01，表示触发要求高。依次类推。

上述方式虽然可以将漏洞描述信息转为形式背景，但考虑到在形式概念分析中概念数量会随着形式背景的增长呈指数倍扩增，为了降低概念的计算量，提高计算效率，对通过漏洞特征划分的 30 个属性进行约简。通过删除漏洞特征评定等级中危害性最轻的选项，并以同类特征均为 0 表示该选项，可以将属性数降低至 20 位，如表 4 所示。则漏洞 X 的形式背景如表 5 所示。

表 4　简化后漏洞特征等级表

序号	漏洞特征	评定等级	序号	漏洞特征	评定等级
1	访问路径	网络、邻接、本地	6	完整性	严重、一般
2	触发要求	低	7	可用性	严重、一般
3	权限需求	无、低	8	被利用成本	低、中
4	交互条件	不需要	9	修复难度	高、中
5	保密性	严重、一般	10	影响范围	高、中、低

表 5　简化后漏洞 X 的形式背景

	1	2	3	4	5	6	7	8	9	10
X	100	0	01	1	10	00	00	10	00	100

简化后的漏洞特征等级删除了每一个漏洞特征评定等级中的最后一项，并以其他项均为 0 表示，例如表 5 中漏洞特征序号为 2 的值为 0，表示触发要求高；漏洞特征序号为 6 对应的值为 00，表示不影响承载信息的完整性。依次类推。

使用简化后漏洞特征等级获得的形式背景可以在不减少漏洞描述信息的情况下快速计算出漏洞概念信息。

2.2　关于漏洞信息的概念格构造

利用漏洞信息作为形式背景，可以构建漏洞相关的概念格，帮助分析人员更深入地了解漏洞之间的关系。本文使用 InClose3 算法[18] 对漏洞信息进行处理，提取其中的概念和属性，并通过相应算法将这些概念和属性之间的关系进行表示和处理，从而构建出漏洞概念格。其中每个节点表示一个概念，节点之间的关系表示概念之间的相似性和包含关系。通过对漏洞概念格的分析，可以帮助发现漏洞之间的共性和差异性，进而为漏洞挖掘、

漏洞预测、漏洞修复等提供有力的支持。InClose3 算法如算法 1 所示。

算法 1　InClose3 算法

输入：基于漏洞特征的形式背景；
输出：形式概念。
(1) //calculate concept
(2) for $j \leftarrow v$ upto do
(3) $M^j \leftarrow N^j$
(4) if $j \notin A$ and $N^j \subseteq A \cap V_j$ then
(5) $W \leftarrow X \cap \{j\}^*$
(6) if $X = W$ then
(7) $A = A \cup \{j\}$
(8) else
(9) if $A \cap V_j = W^{*j}$ then
(10) PutInQueue (W, j)
(11) else
(12) $M^j \leftarrow W^{*j}$
(13) ProcessConcept $((X, A))$
(14) while GetFromQueue (W, j) do
(15) $B \leftarrow A \cup \{j\}$
(16) InClose3 $((W, B), j+1, \{M^v \mid v \in V\})$

InClose3 算法可以将漏洞信息以概念的形式表示，概念的外延表示具有内涵中漏洞特征的全部漏洞，概念的内涵表示概念外延中所有漏洞共同具有的最大特征集合。使用 InClose3 算法虽然可以计算出所有概念，但在形式概念分析中，概念数量随形式背景规模增大呈指数倍增长，当漏洞数量较多时，为了增加个别概念而重新计算所有概念的方式显然效率很低。为此，当漏洞数量较多时，采用概念格增量算法对原有概念格进行改动。

FCboUpdate 概念格增量算法[19]具有不需要读取全部的概念格信息，仅输入形式背景就可以实现增加元素的功能，因此，该算法对内存的消耗更低。本文使用 FCboUpdate 算法实现增加漏洞信息的功能。FCboUpdate 算法具体实现过程如算法 2 所示。

算法 2　FCboUpdate 算法

输入：基于漏洞特征的形式背景；
输出：新更新或修改的概念。
(1) //add or update concept
(2) if new = true then
(3) store concept (A, B) as new
(4) else
(5) store concept (A, B) as modified
(6) if $B = Y_U$ or $y > n_U$ then
(7) return
(8) set min to Y_U
(9) for j from n_U downto y do
(10) set M_j to N_j
(11) set min to min \ $\{j\}$
(12) if $j \notin B$ and $j \in Y_N$ and $N_j \cap Y_{U,j} \subseteq B \cap Y_{U,j}$ then
(13) set C to $A \cap \{j\}^{*\prime}$
(14) set D to $C^{*\prime}$
(15) if $B \cap Y_{U,j} \subseteq D \cap Y_{U,j}$ then
(16) nnew = $((C \cap X)^{*\prime} \neq D)$
(17) if nnew = true or $(C \cap X) \subset C$ then
(18) put $((C, D), \text{nnew}, j+1)$ to queue
(19) if $B \cap \text{min} = D \cap \text{min}$ then
(20) if new = true or nnew = true then
(21) store link form (A, B) to (C, D) as new
(22) set min to min $\cup \{j\}$
(23) set M_j to D
(24) for i from n_U downto y not in Y_N and form $y-2$ downto 0 do
(25) if $i \notin B$ then
(26) set E to $A \cap \{i\}^{*\prime}$
(27) set F to $E^{*\prime}$
(28) set min to min \ $\{i\}$
(29) if $B \cap \text{min} = F \cap \text{min}$ then
(30) if new = true or $(E \cap X)^{*\prime} \neq F$ then
(31) store link from (A, B) to (E, F) as new
(32) set min to min $\cup \{i\}$
(33) While get $((C, D), \text{nnew}, j)$ from queue do
(34) FCboUpdate $((C, D), \text{nnew}, j), \{M_y \mid y \in Y_U\}$

2.3　漏洞分析预处理模块原型设计

为了进一步描述漏洞信息分析预处理模块设计方案，本节提出基于概念格的漏洞分析预处理模块的设计原型。预处理模块主要分为三层，分别是用户界面、逻辑层与数据访问层，层与层之间通过约定的接口进行通信，且禁止跨层之间通信。具体架构图如图 2 所示。

用户界面包括概念格视图和交互选项。概念格视图用于表示概念之间的层级关系，并通过格结构展示出来；交

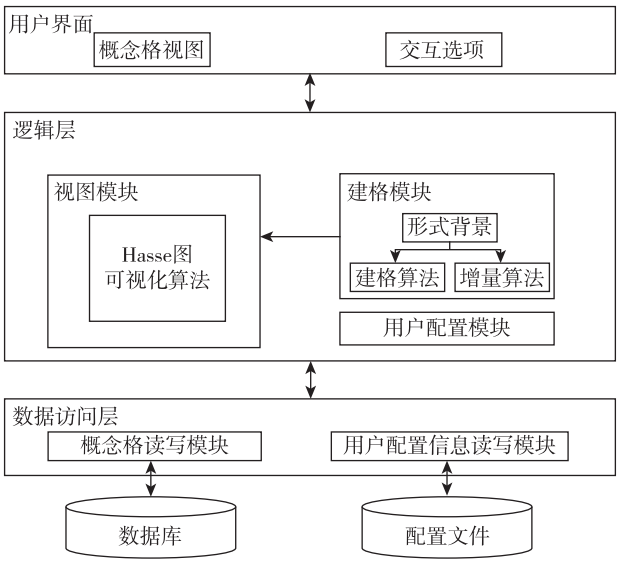

图 2　漏洞分析系统原型设计图

互选项用于进行人机交互，使用者可通过交互选项选择查看某一概念或某些概念所包含的漏洞信息和漏洞特征。

逻辑层用于存储主要执行算法和用户配置信息。建格模块将漏洞信息转为形式背景，并通过建格算法或增量算法构建概念。为了将概念结果通过图形化界面清晰地展示给用户，在逻辑层使用了 Hasse 图可视化算法处理概念。Hasse 图可视化算法可以通过节点表示概念，节点间连线表示概念之间的偏序关系，使用户更加直观地分析不同概念之间的关系。用户配置模块用于处理配置文件，通过不同配置文件可以对概念格中展示的漏洞信息进行处理。

数据访问层是软件系统与外部数据之间的接口，用于读取数据库和配置文件相关的信息。

3　实验结果及分析

根据漏洞特征对例 2 中六种不同漏洞进行聚类分析，快速获得所有漏洞聚类情况。

例 2　表 6 是一个由六种不同漏洞构成的形式背景 (G, M, I)，其中对象集 G 为 a，b，c，d，e 和 f 共六种不同漏洞，M 为简化后的漏洞特征。其中，a 漏洞和 d 漏洞为高危漏洞，e 漏洞和 f 漏洞为中危漏洞，b 漏洞和 c 漏洞为低危漏洞。

表 6　漏洞特征形式背景表

	1	2	3	4	5	6	7	8	9	10
a	100	1	10	0	10	00	00	10	00	100
b	001	1	01	0	01	01	01	10	00	001
c	001	1	00	0	00	10	10	10	00	010
d	100	0	10	1	10	01	10	01	01	100
e	010	1	01	1	01	01	01	01	01	010
f	000	0	10	0	10	10	10	00	10	100

由例 2 中形式背景计算出的概念如表 7 所示，概念的外延为漏洞集合，内涵为漏洞特征的评定等级集合。例如序号为 2 的概念为（{a，b，c，e}，{4}）。根据评定等级位数排列，第 4 位为漏洞特征序号为 2 的第 1 位，即触发要求为低。表示 a，b，c，e 四个漏洞共同具有的漏洞特征为触发要求为低，且同时具触发要求低漏洞特征的所有漏洞为 a，b，c，e 四个漏洞。则在接下来的漏洞分析中，可以从 a，b，c，e 四个漏洞触发要求均为低的特征出发，对这四个漏洞的触发方式进行深入分析。

表 7　漏洞特征形式背景表

序号	概念	序号	概念	序号	概念
1	({G}, {})	8	({b, c}, {3, 4, 14})	15	({d}, {1, 5, 7, 8, 11, 12, 15, 17, 18})
2	({a, b, c, e}, {4})	9	({d, e}, {7, 11, 15, 17})	16	({c, e}, {4, 19})
3	({b, d, e}, {11})	10	({a, d, f}, {5, 8, 18})	17	({c}, {3, 4, 10, 12, 14, 19})
4	({c, d, f}, {12})	11	({a, d}, {1, 5, 8, 18})	18	({e}, {2, 4, 6, 7, 9, 11, 13, 15, 17, 19})
5	({c, f}, {10, 12})	12	({d, f}, {5, 8, 12, 18})	19	({b}, {3, 4, 6, 9, 11, 13, 14, 20})
6	({b, e}, {4, 6, 9, 11, 13})	13	({a}, {1, 4, 5, 8, 14, 18})	20	({}, {M})
7	({a, b, c}, {4, 14})	14	({f}, {5, 8, 10, 12, 16, 18})		

通过表 7 中的数据可以获取漏洞根据其漏洞特征进行聚类的所有情况，相关工作人员可以根据概念进一步从概念内涵中漏洞特征的角度分析概念外延中漏洞的潜在联系，表明方法的准确性与有效性。

由例 2 中形式背景构成的概念格如图 3 所示，节点的序号与表 7 中序号相对应，节点间的连线表示不同概念之间的层级关系，通过不同概念间的层级关系，可以在添加或减少漏洞特征时通过漏洞集合的变化为漏洞分析提供不同的考虑方案。例如序号为 2 的概念是序号为 6 的概念的父节点，序号为 2 的概念为（{a，b，c，e}，{4}），序号为 6 的概念为（{b，e}，{4，6，9，11，13}）。通过这两个概念可以分析出，漏洞 a，b，c，e 均具有属

性 4 表示的漏洞特征，即触发要求低；b，e 漏洞具有属性 4，6，9，11，13 表示的漏洞特征，即触发要求低、权限要求低、保密性等级为一般、完整性等级为一般、可用性一般。在考虑触发要求为低的特征时，会有 a，b，c，e 四个漏洞，而添加一些漏洞特征后，则仅剩 b，e 两个漏洞，在分析时，可以考虑 b，e 两个漏洞与 a，c 漏洞之间的共性与个性以及 b 漏洞和 e 漏洞之间、a 漏洞和 c 漏洞之间的共性与个性。

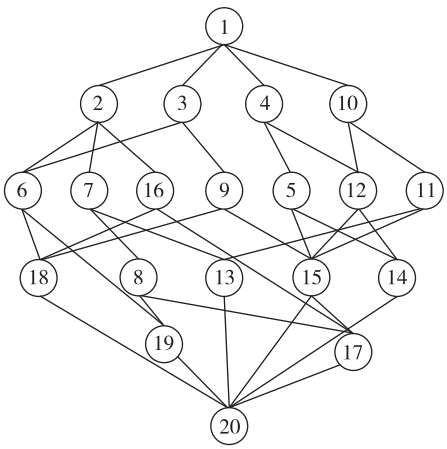

图 3　例 2 中漏洞的概念格图

4　结束语

本文以形式概念分析为基础，将漏洞特征信息转换为形式背景，并构建漏洞特征信息概念格，从而对漏洞进行聚类分析，将具有相同特征的漏洞以概念的形式表示，为漏洞分析管理人员提供参考。文中不仅提供了针对不同情况构建漏洞概念格的方法，也提供了漏洞分析系统设计原型图。最后，通过实验对漏洞进行聚类分析，表明该方法的有效性与准确性。

本文使用漏洞特征对漏洞进行聚类，并将聚类结果返回给用户，这只是漏洞分析中的一个模块。根据漏洞聚类返回的结果对同类的不同漏洞进行深入分析，挖掘不同漏洞特征之间的潜在联系，以及潜在联系产生的原因才是之后关于漏洞分析工作的主要内容。

在后续工作中，将继续完善漏洞分析预处理方法，在每个概念节点中添加更加详细的漏洞信息。工业生产中经常使用漏洞产生原因和漏洞资产等因素对不同漏洞进行分析，为此，后续工作也将探索通过漏洞的类型、产生的原因、来源等信息分析不同漏洞之间的关系，从而为用户提供更加方便的使用体验。

参考文献

[1] 郭娴，杨安，朱丽娜. 工业控制系统信息安全漏洞管理思考与实践 [J]. 中国信息安全，2022（6）：40－43.
[2] 吴世忠，郭涛，董国伟，等. 软件漏洞分析技术进展 [J]. 清华大学学报（自然科学版），2012，52（10）：1309－1319.
[3] 刘剑，苏璞睿，杨珉，等. 软件与网络安全研究综述 [J]. 软件学报，2018，29（1）：42－68.
[4] 吴世忠. 信息安全漏洞分析回顾与展望 [J]. 清华大学学报（自然科学版），2009，49（S2）：2065－2072.
[5] 刘强，殷建平，蔡志平，等. 基于不确定图的网络漏洞分析方法 [J]. 软件学报，2011，22（6）：1398－1412.
[6] 熊杰. 工业控制系统漏洞分析与攻击模拟环境设计与开发 [D]. 武汉：华中科技大学，2017.
[7] 曲海阔. 面向工业联网设备的漏洞分析系统设计与实现 [D]. 哈尔滨：哈尔滨工业大学，2021.
[8] 杨诗雨，苏丽丽，侯元伟，等. 面向漏洞管理的工作流技术应用研究 [J]. 北京理工大学学报，2019，39（9）：967－973.
[9] 信息安全技术 网络安全漏洞分类分级指南（GB/T 30279—2020）[S]. 2020.
[10] 信息安全技术 网络安全漏洞管理规范（GB/T 30276—2020）[S]. 2020.
[11] 信息安全技术 网络安全漏洞标识与描述规范（GB/T 28458—2020）[S]. 2020.
[12] WILLE R. Restructuring lattice theory：an approach based on hierarchies of concepts [M]. Proceedings of the NATO Advanced Study Institute，1982：445－470.
[13] TELCIAN A S，CRISTEA D M，SIMA I. Formal concept analysis for amino acids classification and visualization [J]. Acta Universitatis Sapientiae，Informatica，2020，12（1）. DOI：10.2478/ausi－2020－0002.
[14] RAFAT A M，BLASI A H. Software evolution understanding：automatic extraction of software identifiers map for object-oriented software systems [J]. Journal of Communications Software and Systems，2021，17（1）：20－28.
[15] AROUR K，YEFERNY T. Formal concept analysis based user model for distributed systems [J]. Multimedia Tools and Applications，2017，76（15）：16085－16105.
[16] 刘美玉. 基于三支概念分析的推荐算法研究 [D]. 西安：西安电子科技大学，2021.
[17] 祁建军，魏玲，姚一豫. 三支概念分析与决策 [M]. 北京：科学出版社，2019：34－39.
[18] ANDREW S. A 'Best-of-Breed' approach for designing a fast algorithm for computing fixpoints of Galois Connections[J]. Information Sciences：An International Journal，2015，295（3）：633－649.
[19] OUTRATA J. A lattice-free concept lattice update algorithm based on* CbO [C]//The Tenth International Conference on Concept Lattices and Their Applications，2013：261－274.

（收稿日期：2023－05－15）

作者简介：

王绍杰（1983－），男，硕士，高级工程师，主要研究方向：工控信息安全。

祁斌（1995－），男，硕士研究生，主要研究方向：形式概念分析。

万佳蓉（1996－），女，硕士，工程师，主要研究方向：工控信息安全。

版权声明

凡《网络安全与数据治理》录用的文章，如作者没有关于汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权等版权的特殊声明，即视作该文章署名作者同意将该文章的汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权授予本刊，本刊有权授权本刊合作数据库、合作媒体等合作伙伴使用。同时，本刊支付的稿酬已包含上述使用的费用，特此声明。

《网络安全与数据治理》编辑部