Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.035694
Article

Critical Relation Path Aggregation-Based Industrial Control Component Exploitable Vulnerability Reasoning

Zibo Wang[1,3], Chaobin Huo[2], Yaofang Zhang[1,3], Shengtao Cheng[1,3], Yilu Chen[1,3], Xiaojie Wei[5], Chao Li[4] and Bailing Wang[1,3,*]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Weihai, 264209, China
[2]National Computer System Engineering Research Institute of China, Beijing, 100083, China
[3]School of Cyber Science and Technology, Harbin Institute of Technology, Harbin, 150001, China
[4]Weihai Cyberguard Technologies Co. Ltd., Weihai, 264209, China
[5]Faculty of Science, Vrije Universiteit Amsterdam, Amsterdam, 1081HV, Netherlands
*Corresponding Author: Bailing Wang. Email: wbl@hit.edu.cn
Received: 31 August 2022; Accepted: 26 October 2022

Abstract: With the growing discovery of exposed vulnerabilities in the Industrial Control Components (ICCs), identification of the exploitable ones is urgent for Industrial Control System (ICS) administrators to proactively forecast potential threats. However, it is not a trivial task due to the complexity of the multi-source heterogeneous data and the lack of automatic analysis methods. To address these challenges, we propose an exploitability reasoning method based on the ICC-Vulnerability Knowledge Graph (KG) in which relation paths contain abundant potential evidence to support the reasoning. The reasoning task in this work refers to determining whether a specific relation is valid between an attacker entity and a possible exploitable vulnerability entity with the help of a collective of the critical paths. The proposed method consists of three primary building blocks: KG construction, relation path representation, and query relation reasoning. A security-oriented ontology combines exploit modeling, which provides a guideline for the integration of the scattered knowledge while constructing the KG. We emphasize the role of the aggregation of the attention mechanism in representation learning and ultimate reasoning. In order to acquire a high-quality representation, the entity and relation embeddings take advantage of their local structure and related semantics. Some critical paths are assigned corresponding attentive weights and then they are aggregated for the determination of the query relation validity. In particular, similarity calculation is introduced into a critical path selection algorithm, which improves search and reasoning performance. Meanwhile, the proposed algorithm avoids redundant paths between the given pairs of entities. Experimental results show that the proposed method outperforms the state-of-the-art ones in the aspects of embedding quality and query relation reasoning accuracy.

Keywords: Path-based reasoning; representation learning; attention mechanism; vulnerability knowledge graph; industrial control component

1 Introduction

ICS is a typical critical infrastructure integrating the functions of controlling operations, data acquisition, and process monitoring, which mainly relies on a variety of proprietary components [1]. It is a universal acknowledgement that these components are designed without security in mind because of the isolated physical environment. Nevertheless, current ICS administrators and vendors introduce multiple interconnections and general-purpose configurations, which has broken the original boundaries surrounding the components and posed more potential threats to ICS [2]. After a deep inspection of ICS security incidents occurred in recent decades, one of the root causes for opening up attack surfaces is the exposed vulnerability of ICCs.

Vulnerability management techniques are maturing in conventional Information Technology (IT) systems, while they are not well-implemented in the ICS, especially in its unique Operation Technology (OT) sectors. For example, active scanning and patching have more or less an undesirable impact on the stable and continuous running of ICCs. Performing such tasks frequently is not allowed under the considerations of safety and cost. Additionally, not all discovered vulnerabilities could be exploited by attackers. In most cases, ICS administrators appear to tolerate a few well-known vulnerabilities that always have low exploit possibilities [3]. Therefore, exploitable vulnerability identification is crucial for the ICCs.

The main challenges lie in three aspects. Firstly, large quantities of heterogeneous ICCs are adopted in ICS, which increases not only ICS complexity but also corresponding vulnerability analysis difficulties. In spite of some authoritative vulnerability knowledge bases such as National Vulnerability Database (NVD, https://nvd.nist.gov/), Common Platform Enumeration (CPE), Common Vulnerability Scoring System (CVSS, https://www.first.org/cvss/) and Exploit Database (Exploit-DB, https://www.exploit-db.com/) providing valuable references of security expert experiences, multi-source information is scattered and correlations among them are prone to be neglected by security analysts. Secondly, such knowledge bases are based on fixed assessment methods, not involved in specific scene contexts such as network topologies and security policies, but the contextual information of each ICC is a necessary prerequisite for verifying whether the vulnerability can be exploited or not. Thirdly, it lacks an explicit method to accomplish a series of relevant missions in an automatic manner to aid in obtaining a list of exploitable vulnerabilities [4].

In order to overcome these challenges, domain-specific KGs have been widely incorporated into security analysis, which links the huge and multiple types of data mentioned above. Moreover, the KG organizes a collective of security entities and captures the semantically-interconnected relations among them according to a customized and conceptual ontology [5]. What is more, the underlying structure of the constructed KG, depicted as a labeled graph, effectively underpins knowledge representations and inferences for implicit relations. As a result, identifying exploitable vulnerabilities in the ICCs is further made possible by means of reasoning a query relation between a specific pair of entities based on the KG pertaining to security expert experiences and scene context information.

Mainstream KG-based reasoning methods focus less on relation paths between a pair of entities, but the paths contain sufficient potential evidence for exploitable vulnerability identification. In the limited amount of relation path-based literature, most current methods only consider one of the multiple paths as a unique feature to perform reasoning tasks [6,7], which is not suitable for this work. The evidence of exploitable vulnerabilities is distributed in several paths not only one. Each path possesses a different level of impact in terms of final reasoning. Besides, there are also irrelevant paths between a pair of entities. Although some methods also account for paths with different levels

to infer the relation [8,9], they do not differentiate paths based on relevance and directly utilize all of them, which is unconducive for the calculation performance with the increasing scale of the KG.

This work aims to bridge the aforementioned gaps by proposing a critical relation path aggregation-based reasoning method for identifying exploitable vulnerabilities. First and foremost, under a guideline of ontology designed from the point of view of attackers, an ICC-Vulnerability KG is constructed with the integration of multi-source data dispersed in vulnerability knowledge bases and scene context information. Subsequently, the KG is embedded into a continuous low-dimensional space where entities and relations are represented as vectors. The relation path vector is obtained by accumulating all relation vectors on each path between a pair of entities, facilitating the next reasoning task. Finally, a critical path selection algorithm on the basis of similarity increment with the query relation vector is introduced into the reasoning method. And then multiple critical paths are aggregated to collectively determine whether the query relation between a specific pair of entities is valid, thereby identifying exploitable vulnerabilities. Note that these two aggregations in the proposed method are realized with the help of the attention mechanism. One is for aggregating initial vectors with semantic and structural information as well as adjacent vectors of entities and relations, and the other is for aggregating critical paths that are assigned attentive weights.

The contribution of this paper is summarized as follows:

(C1) Compared with existing vulnerability KGs, we highlight relevant concepts regarding exploitability while devising a security ontology that guides the construction of the ICC-Vulnerability KG.

(C2) The aggregation effect of the attention mechanism is significantly exerted in both representation learning for the KG and the combination of relation paths that indicate potential evidence.

(C3) Redundant relation paths are effectively avoided by virtue of similarity increment with the query relation vector in the selection algorithm, which enhances the performance of reasoning tasks.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 provides an overview of the proposed method. In Section 4, an ICC-Vulnerability KG is constructed according to a designed ontology. A vector representation for relation paths between a pair of entities is given in Section 5. In Section 6, the reasoning for the query relation with respect to the exploitability of each vulnerability is elaborated on the basis of relation path aggregation. Experimental results are demonstrated in Section 7. At last, the whole paper is concluded, followed by future research directions.

2 Related Work

In this section, we briefly review the related literature on the domain-specific KGs and reasoning-based analysis that emerged in the past five years. Security-oriented ontologies as well as KGs are designed and implemented in respective application scenarios. After that, we introduce a reasoning-based security analysis on the prediction or assessment of vulnerabilities and their exploitability.

2.1 Security Domain-Specific KGs

Security KGs have two distinct characteristics, namely connections and reasoning between different entities. Linking scattered and heterogeneous security knowledge with unified specifications and standards, KGs are treated as cross-domain and large-scale databases. To that end, numbers of researchers make more efforts in the representation, extraction, and storage of knowledge. In terms of the reasoning function, KGs are primary sources of missing relationships complement and new facts

mining according to the existing security knowledge. Aiming to construct an ICC-Vulnerability KG that is appropriate for the performance of reasoning tasks, security-oriented ontologies and KGs in preceding literature provide a wealth of inspiration.

A general ontology for security assessment comprises basic concepts, i.e., assets, vulnerabilities, and attacks [10]. Various security ontological models are expanded with conceptualizations of diverse information. For example, Wang et al. developed an ontology regarding social engineering attacks, including eleven core entity types and twenty-two relevant relations [11]. For the same purpose of automatically identifying security risks in the ICS, an ontology presented by Eckhart et al. combined with a transformation from the Automation Markup Language (Automation-ML) to Web Ontology Language (OWL) [5], and a hybrid ontology proposed by Alanen et al. harmonized concepts among safety, security, and dependability on the basis of current industry standards to assist in the threat analysis [12]. Just as we are concerned with the vulnerability domain, Syed conceptualized an ontological representation that integrates social media intelligence with official information for the purpose of vulnerability management [13]. Du et al. developed a software vulnerability ontology to keep track of links between vulnerabilities and software components, applying two ontology matching techniques [14].

In contrast with the above ontologies focusing on high-level abstraction, security KGs place more emphasis on easy accessibility and significant comprehensibility for specific data in practice. Automated extraction of entities and relations plays a vital role in KG construction. Sarhan and Spruit proposed an attention-based open information extraction method to get fact triples from unstructured Advanced Persistent Threat (APT) reports. Meanwhile, they labeled these triples in conjunction with a neural NER [15]. Considering the relation extraction, Shen et al. provided an ICS data-driven framework with a security KG [16]. A convolutional neural network associated deep residuals with a multi-instance attention mechanism to avoid the impact of noise data. On the other side, an ongoing research project, SEPSES, maintains a KG with the integration of up-to-date instance data from both publicly and locally available information [4]. In order to address the problems of the existing Common Vulnerabilities and Exposures (CVE, https://cve.mitre.org/) in readability and visualization, the Neo4j (https://neo4j.com/) graph database is employed to construct a vulnerability KG, which is effective for security data analysis in an intuitive way [17].

2.2 Reasoning-Based Security Analysis

Either the ontology or the KG has the capability of independent accomplishment of reasoning tasks in a specific domain. For ontology-based reasoning methods, Semantic Web Rule Language (SWRL) rules are adopted for relational inferences. Wu et al. expressed the relationships of attack scenarios in the SWRL rules and assessed potential threats by inferring vulnerabilities and their induced attacks [10]. Similarly, Zhang et al. revealed the implicit relation based on the inference rules to discover vulnerable platforms in the IoT environments [18]. To some extent, ontology-based reasoning methods cannot meet the demands of cost computing and rule generation complexity when instances increase. An original intention of KG-based methods is to excavate implicit relationships from more instances. By calculating the conditional probability of a pair of weaknesses belonging to the same product entity in a vulnerability KG, Qin et al. mined hidden weakness chains of compromised products in a statistical way [19].

In addition, embedding entities and relations into a continuous vector space is a feasible solution to massive ones. Symbolic entities and relations in the KGs are represented as vectors, which improves the computing performance in reasoning tasks. At the same time, relevant structural and descriptive

information can also be embedded into these vectors to enhance the accuracy of reasoning results [20–22]. Han et al. introduced the description of each Common Weakness Enumeration (CWE, https://cwe.mitre.org/) into a translation-based representation learning, constructing a semantic vector space of the KG [20]. The generated embeddings of entities and relations provide a foundation for reasoning about the CWE links and their common consequences. Similarly, Xiao et al. reasoned a series of within-type and across-type software security-oriented relationships among different databases, including CVE, CWE, as well as Common Attack Pattern Enumeration and Classification (CAPEC, https://capec.mitre.org/) [21]. Independent treatments of each triple in KGs in the translation-based methods are unreasonable due to the neglect of the rich information existing in the neighbors. To cope with that problem, Yuan et al. presented a semantic text-enhanced graph attention network model for reasoning relations of security entities [22].

Models using such embeddings are suitable for reasoning about a direct relation between a pair of entities. In other words, they could infer new triples based on known complete triples. Simultaneously, paths that are formed by a sequence of relations are also meaningful for those similar inferences. Jia et al. applied a Path Ranking Algorithm (PRA) for reasoning relations [7]. A set of relation paths between a pair of entities are collected by random walks [23]. After that, a binary classification with each discovered path is carried out to determine whether the query relation is valid. There are two obvious drawbacks to the PRA-based reasoning method. One is for unsatisfactory calculating performance as the number of paths increases, and the other is for the inaccuracy of reasoning results owing to leveraging each path equally. However, the security KG-based reasoning mentioned in this subsection focuses little on the exploitable vulnerabilities attracting a majority of attackers.

In summary, neither the existing security ontology models nor their related KGs are suitable for the task in this work because they ignore the modeling of exploit behavior on vulnerabilities for the ICCs from the point of view of potential attackers. Moreover, ongoing research interests in path-based reasoning lie in representation learning with semantics [24] and the capture of fine-grained features with the help of the attention mechanism [25]. However, redundant relation paths between a given pair of entities limit the performance of the reasoning tasks. To the best of our knowledge, the proposed model first introduces similarity calculation into the reasoning to select critical relation paths, which improves the performance compared with existing attention-based methods.

3 Proposed Method

The ultimate goal of the proposed method is the identification of exploitable vulnerabilities in the ICCs. We perform reasoning tasks in a predefined ICC-Vulnerability KG to achieve that goal. Choosing the domain-specific KG as our research backbone meets three demands of reasoning tasks, including integrations of scattered security data, representations of relevant knowledge, and combinations of multiple pieces of evidence. More specifically, constructing KG relies on the conceptualization of the security ontology in line with the practical exploit scenarios of the ICC domain, and embeddings of entities and relations facilitate the representation of relation paths. In particular, we emphasize the reasoning using critical relation path aggregations that contain multiple core evidence of an exploitable vulnerability. To that end, the proposed method is separated into three building blocks, as shown in Fig. 1.

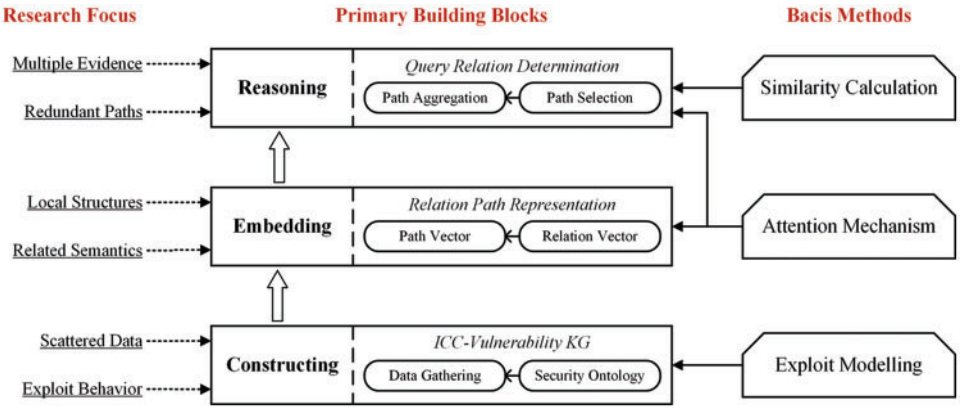

Figure 1: Overview of the proposed method

(B1) ICC-Vulnerability KG Construction: From the perspective of attackers, a security-oriented ontology is developed with exploit modeling that mainly concerns pre-conditions and post-conditions of a given vulnerability, i.e., privileges and connectivity. The entity types and relations in the ontological model reflect a collection of exploit behavior on vulnerabilities. In the model, it is of importance to refine the concept of "Exploitable Vulnerability" and distinguish it from "Common Vulnerability". The construction of the ICC-Vulnerability KG is in accordance with the ontology, organizing scattered security data in a formal and structured format. Such data consists of expertise in security vulnerability databases and heterogeneous information in the ICC scene context. Thus, the domain-specific KG is primary and substantial for subsequent representation and reasoning to identify exploitable vulnerabilities.

(B2) Relation Path Representation: We take account of relation paths that imply evidence for the identification of exploitable vulnerabilities. These paths are embedded in a low-dimensional space and represented as vectors. Each path is constituted by a sequence of relations between a pair of entities. Hence, the path vector in this work is directly obtained by accumulating related relation vectors as well. In other words, the representation quality of each path vector relies on that of each relation vector. There are two steps for the improvement of embedding regarding related semantics and local structures. An initial vector of each entity or relation is generated by a semantic embedding of the security corpus and a translation-based structural embedding. Furthermore, the attention mechanism is introduced for aggregating vectors of adjacent entities and relations that surround a given entity. Accordingly, it is convenient for processing corresponding vectors to select paths and aggregate them for reasoning.

(B3) Critical Path Aggregation and Reasoning: KG-based reasoning in this work concretely refers to determining whether the query relation "can exploit" exists between the initial entity "Attacker" and the target entity "Exploitable Vulnerability". Relation paths between the pair of entities contain potential evidence to support the reasoning task, but some of them seem to be redundant. We define "Critical paths" that have a strong correlation with the ultimate reasoning goal. On the basis of two practical observations, a search algorithm with vector similarity calculation is proposed for the selection of the critical paths. Critical paths do not contribute equally to reasoning, and parts of them need to be applied together for a determination of query relation. Hence, we introduce the attention mechanism again, assigning a distinguishable weight to each path and subsequently aggregating relevant paths. After that, an aggregative path vector between the initial entity and the target entity is

generated with respect to the core evidence of the reasoning goal. Finally, we employ a binary classifier for each aggregative vector to determine if the query relation is valid.

4 ICC-Vulnerability Knowledge Graph Construction

In this section, we briefly describe the workflow of the ICC-Vulnerability KG construction. An ontological model is built by introducing exploit modeling that focuses on attacker-oriented pre-conditions and post-conditions of vulnerabilities, which underlines the conceptualization of the exploit behavior.

The ICC-Vulnerability KG is constructed in a top-down manner. It consists of a pattern layer and a data layer. The pattern layer depends on the ICC-Vulnerability ontology to account for concepts like asset information, scenario conduction, and weakness knowledge. The data layer covers the multi-source data in structured, semi-structured, and unstructured forms. The pattern layer provides a schema for the data layer and a guideline for knowledge extraction and fusion. The knowledge is stored as multiple fact triples after the quality evaluation. In this work, we utilize a rule-based method to extract entities, relations, and properties from the convergence of those different forms. The workflow of the ICC-Vulnerability KG construction is illustrated in Fig. 2.

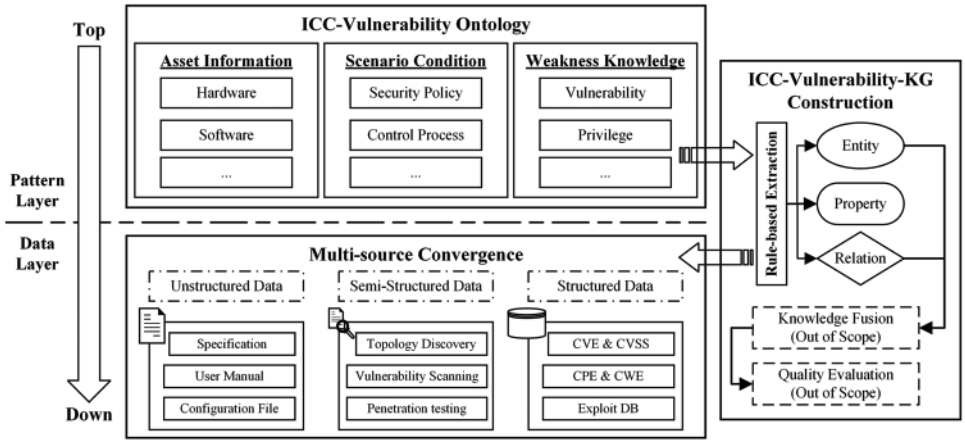

Figure 2: Workflow of the ICC-Vulnerability KG construction

As a core of the security-oriented KG, the ICC-Vulnerability Ontological Model in Fig. 3 involves eight entity types and twelve relations. The model is built around proprietary and general-purpose components in typical industrial control scenarios, including various devices, Operating Systems (OSs), and application software. These three concepts, including *Device*, *Operating System*, and *Application Software*, fall under the category of asset information. Relevant knowledge can be extracted from specifications, user manuals, configuration files, and analytical reports of asset management tools.

Vulnerabilities that breach the security policies have probably been discovered in the components mentioned above. To perform the reasoning task in this work, we define two types of concepts regarding vulnerability, namely "Common Vulnerability" (*C-Vul.*) and "Exploitable Vulnerability" (*E-Vul.*). The *C-Vul.* refers to a vulnerability in the security database, such as NVD and CVE, and a vulnerability may correspond to several similar compromised components. For instance, a vulnerability assigned a CVE identifier is disclosed for different versions of products from the same

vendors. The *E-Vul.* concept tightly corresponds to a unique component running in a specific industrial environment. Meanwhile, the *E-Vul.* has seamless integration with the concepts of *Privilege* and *Connectivity* that gain constant attention from attackers.

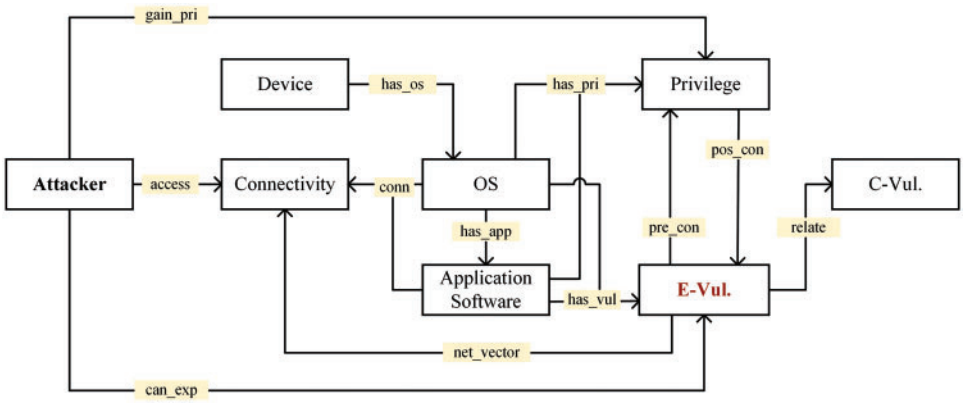

Figure 3: ICC-Vulnerability ontological model

Incorporating the ideas of pre-conditions and post-conditions in the exploit modeling [26], we leverage two concepts, *Connectivity* and *Privilege*, whose descriptions of the exploit behavior can be treated as the reasoning conditions for identifications of exploitable vulnerabilities. *Connectivity* involves the reachability of the network topology and the scenario that accesses the vulnerability. *Privilege* is defined as operation permissions and accessible scope performed by users at different levels. Particularly, attackers could exploit certain vulnerabilities to change the existing privileges of compromised components. These four concepts, including the *C-Vul.*, the *E-Vul.*, the *Connectivity*, and the *Privilege*, all fall under the category of scenario condition and weakness knowledge. Relevant knowledge can be extracted from the reports of vulnerability scanning and penetration testing, as well as the information in well-known security databases.

In order to cooperate with the above concepts, we bring in the Attacker concept and three relevant relations, including "*access*", "gain privilege" (*gain_pri*), and "can exploit" (*can_exp*). These three relations demonstrate three capabilities that potential attackers gradually possess for ICC infiltration. For example, exploiting certain remotely accessible vulnerabilities allows attackers to gain the privileges of a compromised component in the ICS.

In addition to entities and relations, properties that show knowledge in detail have significance for the ontological model. Table 1 shows the properties of the entities mentioned above. The Unique Identifier (Uid) is used to distinguish different entities. The reason why the entity type Attacker only has the Uid is that the key information has been assigned to other types. Its appearance in the model is intuitive for the comprehension of the reasoning objective.

Table 1: Properties for each entity type

No.	Entity type	Property
1	Device	Uid, Product name, Vendor, CPE, Version, Serial No., Central processing Unit (CPU), Detail description

(Continued)

Table 1: Continued

No.	Entity type	Property
2	Operating system	Uid, Product name, Vendor, CPE, Version, Configuration, Detail description
3	Application software	Uid, Product name, Vendor, CPE, Version, Support platform, Detail description
4	Connectivity	Uid, Internet protocol (IP) address, Port
5	Privilege	Uid, Level, Scope
6	E-Vul.	Uid, Vulnerability No., Pre-condition, Post-condition
7	C-Vul.	Uid, CVSS3.0_score, Attack vector, Attack complexity, Authentication, Affected products
8	Attacker	Uid

We have two extra considerations while constructing the KG in this work. Each relation between a pair of concepts is depicted unidirectionally, and its direction can also be presented in a reversing way. It is analogous to describing such concepts in active or passive form, which expresses the same meaning. The reverse relations are conducive to reasoning, especially in the relation path-based approach. We make full use of reverse relations in this work. Furthermore, we collect some keywords in the extraction of entities and relations to form a security corpus applied for enhancements of embeddings.

5 Relation Path Representation

For the sake of the elaboration of reasoning in the following parts, we give a formal description of the KG in this section. And then an attention mechanism embedding approach is introduced to generate high-quality vectors of entities and relations. The relation path is further represented by accumulating a sequence of relation vectors.

Recently, the Graph Attention Network (GAT) has been proven to be a notable success in both embedding and reasoning [8,9,15,16,22]. It enables the attention mechanism to concentrate on the most related parts and aggregates multiple features of triples or relation paths in their respective models. We employ the GAT to embed semantic and structural information into vectors of entities and relations.

The ICC-Vulnerability KG is denoted as $\mathrm{G} = (\mathrm{E}, \mathrm{R})$ and includes a set of tiples T, where E is a set of entities and R is a set of relations. Each tiple (e_h, r, e_t) is made up of the head entity e_h, tail entity e_t, and relation r between them, where $e_h, e_t \in \mathrm{E}$ and $r \in \mathrm{R}$. Especially, a query relation between a pair of given entities is denoted as δ. Each relation path is represented as a sequence of relations $\pi_i = \{r_1, r_2, \cdots r_n\} \in \Pi$, where Π denotes the path set.

Inspired by the existing work [20–22], the initialization vectors that are the input of the GAT are generated by the semantic and structural embeddings. The semantic embedding is to deal with the security corpus of entities or relations by the Word2Vec model, and it obtains the average vectors of a collective of corresponding keywords, where $\mathbf{e}_i^c$ and $\mathbf{r}_i^c$ denote the semantic vectors of the entity and the relation, respectively. The structural embedding is realized by the TransE model [27], where $\mathbf{e}_i^s$ and $\mathbf{r}_i^s$ denote the semantic vectors of the entity and the relation. The initialization vectors of entity e_i and relation r_i are denoted as

$$\begin{cases} \mathbf{e}_i = \mathbf{e}_i^s \oplus \mathbf{e}_i^c \\ \mathbf{r}_i = \mathbf{r}_i^s \oplus \mathbf{r}_i^c \end{cases} \tag{1}$$

where the symbol $\oplus$ is concatenating operation.

Next, given an entity in the KG, the adjacent entity and relation are taken into consideration for generating new embeddings that aggregate more information from the graph structure [28]. The general process of embedding with the GAT is illustrated in Fig. 4. Each layer of the model needs two embeddings as input, namely the entity embedding and the relation embedding. A triple vector is formed by concatenating the vectors of the given entity, its adjacent entity, and the relation. The triple vector is mapped into a higher-dimensional feature space by a linear transformation matrix, obtaining representation vectors of the head entity, the tail entity, and the relation between them. With the application of the single-layer feedforward neural network and the Leaky Rectified Linear Unit (LeakyReLU) non-linear function, an absolute attention parameter is obtained for the triple. Using the Softmax function, the attentive weight is calculated by normalizing all absolute attention parameters of all triples in the neighborhood of the given entity. The new embedding of the entity is generated by accumulating each triple vector assigned the corresponding attentive weight. Following a similar process, a new embedding of relations can be generated. A score function regarding these new embeddings is established for computing training loss.

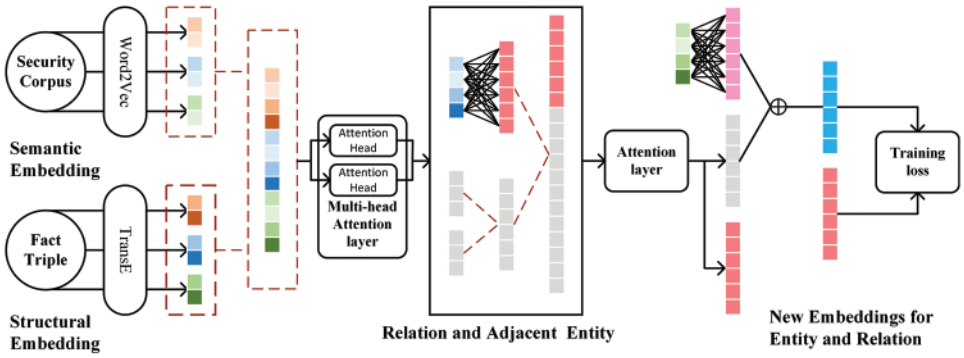

Figure 4: Both semantic and structural embeddings with the attention mechanism

Moreover, the initial embeddings of entities are lost in the above process. Hence, the new embedding of the entity also needs to integrate the embedding generated by the attention layer with the initial embedding. Note that the multi-head attention layer enables the model to capture a wealth of features from the neighborhood, and then the attention layer makes the learning process stable [28].

Summing up all relations that belong to π_i, the path vector π_i is represented as

$$\pi_i = \sum_{k=1}^{n} \mathbf{r}_k \tag{2}$$

6 Query Relation Reasoning

In this section, we first describe the reasoning objective in the KG for the identification of the ICC exploitable vulnerability. After that, a motivating example illustrates the effectiveness of evidence in the relation paths that assist in performing reasoning tasks. Subsequently, a depth-first search algorithm

with a similarity calculation is given to select a collective of critical paths. Finally, we aggregate these critical paths with the help of the attention mechanism and then complete the reasoning tasks.

From the perspective of attackers, a primary concern is whether a vulnerability in the specific scenario could be exploited once some conditions are met. To that end, the ontological model captures what attackers are concerned about, and the KG consolidates multi-source data on the vulnerabilities and their scene context, which provides strong support for the reasoning. Therefore, the identification of exploitable vulnerabilities can be implemented by inferring a relation between specific entity pairs. Specifically, the reasoning objective in this work is to determine whether a query *can_exp* relation is valid between each pair of the *attacker* entity and the *E-Vul.* entity.

6.1 Critical Path Selection

The path that consists of a sequence of relations between the attacker entity and the *E-Vul.* entity contains evidence to determine the *can_exp* relation. We demonstrate a part of the ICC-Vulnerability KG with respect to a specific exploitable vulnerability by a motivating example. The vulnerability corresponds to a common vulnerability that has been disclosed by the CVE and assigned a unique identifier, CVE-2021-20676. The compromised component is a Web User Interface (WebUI) running on the DL8, which is a remote management device. Exploiting such vulnerability allows a remote authenticated attacker to escalate the privilege. The attacker may manipulate the ICS devices connected to the DL8 [29].

As shown in Fig. 5, we select four potential relation paths (some irreverent entities are omitted) and illustrate how to leverage evidence in the paths to determine the *can_exp* relation. Path ① indicates the attacker can gain the App_Admin level privilege by exploiting the vulnerability. Path ② indicates the attacker can have remote access to the exploitable vulnerability via the network. Path ③ indicates the vulnerability exists in the WebUI running on the operating system with version DL8-a_firmware 3.0. It is impossible to directly determine the query relation by utilizing any path alone. The combination of paths ① and ② has favorable evidence for drawing the conclusion that the attacker can exploit the vulnerability. The combination of paths ① and ③ can also draw the same conclusion, but it is less intuitive than the combination of paths ① and ②. The combination of paths ② and ③ is not used for the determination of the query relation. Besides, path ④ indicates the attacker can gain the OS_User level privilege of the operating system. Even if path ④ can also reach the *E-Vul.* entity in the KG, it is redundant for the determination.

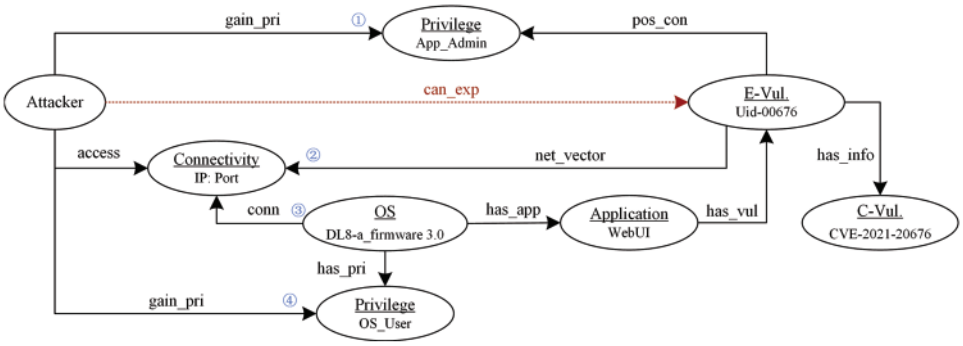

Figure 5: Motivating example for four paths between a pair of entities

According to the analysis of the motivating example, there are two crucial observations that guide the favorable evidence discovery to determine the query relation. (1) Not all relation paths contribute

to determining the query relation. Redundant relation paths exist between the given pair of entities. That is to say, critical relation paths that contain a wealth of evidence are supposed to be selected from all paths for reasoning. (2) Each relation path is of different importance to the reasoning goal. Aggregating some critical relation paths is allowed to complete the reasoning tasks. The former is the foundation for designing the algorithm in this subsection. The latter will be discussed in the next subsection.

The importance of each path depends on the collection of relations that constitute the path. Given the first observation, we select the critical relation paths based on the similarity between each path vector and the query vector. The critical relation path vector is more similar to the query relation vector. The similarity between a relation path vector $\boldsymbol{\pi}_i$ and the query relation vector $\boldsymbol{\delta}$ is defined as

$$\text{sim}\,(\pi_i, \delta) = \frac{\boldsymbol{\pi}_i \cdot \boldsymbol{\delta}}{||\boldsymbol{\pi}_i||\,||\boldsymbol{\delta}||} \tag{3}$$

We propose a critical relation path depth-first search algorithm with similarity calculation. As the search proceeds in the depth direction, a new relation vector is continuously added to the current relation path vector. If the similarity between the current relation path vector and the query relation vector is always monotonically increasing until search stops, the current path is selected as a relation-critical path. In other words, the path will be redundant and removed once the similarity decreases. Furthermore, the path length is taken into consideration for the enhancement of search efficiency. According to the motivating example, a relatively short path contains direct evidence and is more likely to be selected as a critical path. More details are shown in the pseudocode of Algorithm 1. The input of the algorithm includes the target entity that represents a potentially exploitable vulnerability, the maximum path length which is the number of relations, and the query relation vector. The output is the critical path set. The algorithm is implemented by recursive calls. The algorithm backtracks if the similarity in the current step is less than that in the previous step.

Algorithm 1: Critical relation path depth-first search with the similarity calculation

Input: target entity e_t, maximum path length *max_p_len*, query relation vector *q_r_vec*
Output: critical path set *cri_p_set*

(1) **function** DFS_SIM (*cur_rela, cur_ent, cur_p_rela, cur_p_ent, cur_p_vec, last_sim, last_len,* e_t*, max_p_len, q_r_vec, cri_p_set*)
(2) append current relation *cur_rela* to the list of current path relation *cur_p_rela*
(3) append current entity *cur_ent* to the list of current path entity list *cur_p_ent*
(4) get the vector of *cur_rela* and record it as *cur_rela_vec*
(5) *cur_p_vec* ← *cur_p_vec* + *cur_rela_vec*
(6) calculate the length of *cur_p_vec* and record it as *cur_len*
(7) calculate the similarity *cur_sim* between *cur_p_vec* and *q_r_vec* as described in Eq. (3)
(8) **if** *cur_ent* is e_t **then**
(9) add *cur_p_rela* into *cri_p_set*
(10) remove the last elements from *cur_p_rela* and *cur_p_ent*
(11) **end if**
(12) **if** *cur_len* := *max_p_len* **and** *cur_ent* is not e_t **then**
(13) remove the last elements from *cur_p_rela* and *cur_p_ent*
(14) **end if**
(15) **if** *cur_sim* > *last_sim* **and** cur_len > last_len **then**

(Continued)

Algorithm 1: Continued

```
(16)    remove the last elements from cur_p_rela and cur_p_ent
(17)   end if
(18)   for each entity adj_ent and relation adj_rela adjacent to cur_ent do
(19)    if adj_ent is not in cur_p_ent then
(20)     DFS_SIM (adj_rela, adj_ent, cur_p_rela, cur_p_ent, p_vec, cur_sim, cur_len, e_t, max_p_len,
          q_r_vec, cri_p_set)
(21)    end if
(22)   end for
(23)   remove the last elements from cur_p_rela and cur_p_ent
(24)  end function
```

6.2 Critical Path Aggregation-Based Reasoning

To some extent, our reasoning task can be attributed to a link prediction problem. KG embedding is a general solution to that problem. A triple-based score captures the local information function, which is the process for the representation learning of entities and relations. It is not suitable for the reasoning task that relies on multiple relation paths from a global perspective. Although a model such as PTransE [6] integrates some path features into the embedding, it still focuses on the local representation since it continues the score function of the TransE [27]. As a result, we present a critical path aggregation-based reasoning method.

The unique characteristics of the attention mechanism meet two demands from the second observation mentioned in the previous subsection. On the one hand, the query relation can be reasoned by considering multiple critical relation paths collectively. On the other hand, the attentive weight of an individual path is assigned in an automatic and dynamic manner. Aggregation of critical relation paths makes it possible to alternatively extract evidence in order to determine the validity of a query relation. The progress of the reasoning with the critical relation path aggregation is shown in Fig. 6. The principle of the attention mechanism has been broadly elaborated in Section 5. Both the critical relation paths and the query relation are mapped into a higher-dimensional feature space. Calculate the attentive weight of an individual critical path π_i as follows:

$$\alpha_i^{\delta} = \frac{\exp\left[\text{LeakyReLU}\left(\boldsymbol{a}^T\left[\boldsymbol{W}\boldsymbol{\pi}_i \oplus \boldsymbol{W}\boldsymbol{\delta}\right]\right)\right]}{\sum_{k=1}^{n}\exp\left\{\text{LeakyReLU}[\boldsymbol{a}^T\left(\boldsymbol{W}\boldsymbol{\pi}_k \oplus \boldsymbol{W}\boldsymbol{\delta}\right)]\right\}} \tag{4}$$

where $\boldsymbol{W}$ and $\boldsymbol{a}$ are the projection matrix and weight matrix of the single layer feedforward neural network, respectively.

The aggregative path vector between e_h and e_t is represented as

$$\mathbf{ap}^{\delta}\left(e_h, e_t\right) = \text{ELU}\left(\sum_{i=1}^{n}\alpha_i^{\delta}\boldsymbol{\pi}_i\right) \tag{5}$$

where the Exponential Linear Unit is denoted as ELU. The aggregative path vector is passed through a fully-connected layer, which is equivalent to a binary classification of the path vector. The probability vector that determines whether the query relation is valid is given:

$$\mathbf{P}(\delta|e_h, e_t) = \text{softmax}(\mathbf{ap}^{\delta}\boldsymbol{W}_{\text{fc}}) \tag{6}$$

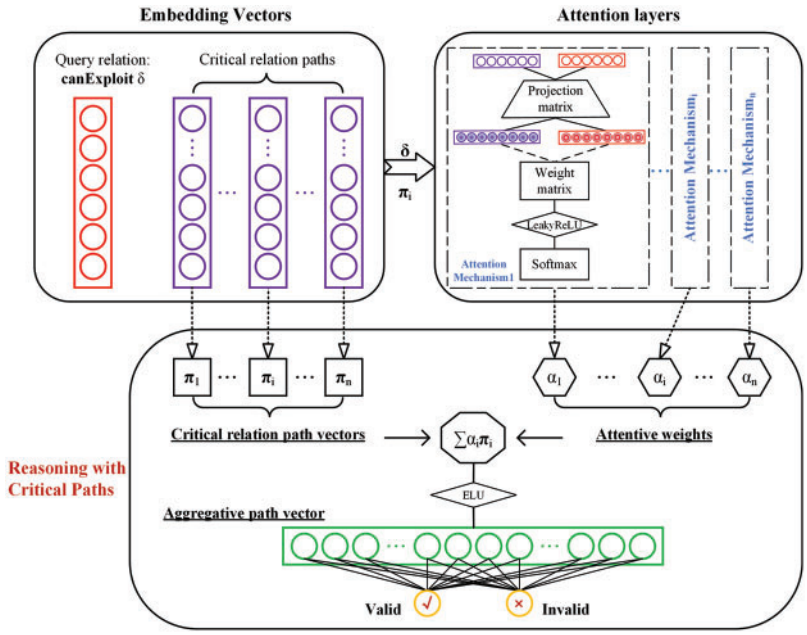

Figure 6: Reasoning with the critical relation path aggregation

where $\boldsymbol{W}_{\text{fc}}$ denotes a weight matrix of a fully connected layer.

The model is trained by a cross-entropy loss function L. The function is defined as follows:

$$L = -\sum_{(e_h,e_t)\in \text{D}^+} \log(\text{P}) - \sum_{(e'_h,e'_t)\in \text{D}^-} \log(1-\text{P}) \tag{7}$$

where P denotes the specific probability regarding the query relation; D^+ and D^- are the positive and the negative sets, respectively.

7 Experimental Results

In Subsection 7.1, we first generate an ICC-vulnerability KG that provides data for embeddings and query relation reasoning. Subsequently, a link prediction experiment is conducted to evaluate the performance of entity and relation embeddings in Subsection 7.2. Finally, we show the effectiveness of Algorithm 1 in selecting a collective of critical relation paths and give primary results on the *can_exp* relation reasoning compared with typical path-based methods. The proposed method is written in Python (Version 3.8.10) and runs on a Linux Centos 7.2 server equipped with an Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40 GHz, an NVIDIA Tesla P100 GPU, and 12 GB of RAM.

7.1 ICC-Vulnerability KG Generation

The KG in the experiment covers the component information of seven mainstream ICS vendors, including Siemens, Mitsubishi, Omron, Emerson, Rockwell, Schneider, and Asea Brown Boveri Ltd. (ABB). Meanwhile, it incorporates a number of general-purpose components from the Microsoft Corporation, which is widely used in the ICS. We utilize keywords that combine the names of vendors and components to collect entities and their properties in the following data sources. The vulnerability knowledge derives from NVD and CVE. The knowledge of scene context is divided into two parts,

namely, information from the configuration of a simulated industrial scenario with some potentially vulnerable components and Industrial Control Systems Cyber Emergency Response Team Advisories (ICSAs, https://www.cisa.gov/uscert/ics/advisories). The simulated industrial scenario is constructed by referring to a large Supervisory Control and Data Acquisition (SCADA) architecture [1]. The advisories timely summarize security issues, vulnerabilities, and exploits existing in the components of a variety of vendors. The statistics of entities in the ICC-Vulnerability KG are shown in Table 2.

Table 2: Statistics of entities in the ICC-Vulnerability KG

Entity type	Device	OS	Application software	Connectivity	Privilege	E-Vul.	C-Vul.
Number	500	550	660	1186	3558	4017	755

Some basic relations are also obtained through the extraction of the entities. Given the three relations of the attacker entity type, we assign some hypothetical initial conditions to the potential attacker and establish rules to add relations between the attacker entity and its connected entities, which imitates the manual assessment of the whole ICS by security analysts. For the sake of verifying the proposed method, we generate a testing dataset. In practice, it is a costly and labor-intensive endeavor, especially for a large-scale ICS. That is why we proposed an automatic reasoning method for the identification of exploitable vulnerabilities. The statistics of relations in the ICC-Vulnerability KG are shown in Table 3. The whole KG contains 11203 entities and forms 54842 triples, including ones with reverse relations.

Table 3: Statistics of relations in the ICC-Vulnerability KG

Index	Relation name	Number
1	assess	1186
2	can_exp	2254
3	gain_pri	1796
4	relate	4017
5	has_app	636
6	conn	1186
7	has_os	500
8	pos_con	1787
9	pre_con	4017
10	has_vul	3558
11	net_vector	4017

7.2 KG Embedding Evaluation

The KG embedding is evaluated by means of link prediction, which refers to predicting the missing entities or relations for incomplete triples. The link prediction tasks consist of an entity prediction and a relation prediction. The detailed evaluation protocol can be found in the embedding-related literature such as [28]. The performance of link prediction is evaluated by three metrics, namely the proportion of correct entities in the top k ranks ($Hits@k$), mean rank (MR), and mean reciprocal rank (MRR) which are

$$Hits@k = \frac{1}{|S|}\sum_{i=1}^{|S|} \mathrm{I}(rank_i \leq k),$$

$$MR = \frac{1}{|S|}\sum_{i=1}^{|S|} rank_i,$$

$$MRR = \frac{1}{|S|}\sum_{i=1}^{|S|} \frac{1}{rank_i}, \tag{8}$$

where $rank_i$ is the link prediction ranking result of the $i-th$ triple; $I(\cdot)$ is a binary function and the value is 1 if the condition $rank_i \leq k$ is true, otherwise the value is 0; S denotes a collective of positive triples in the test set.

Due to the existence of the triples with the reverse relations, the dataset mentioned in the previous subsection cannot be divided in a random way, which intends to avoid a data leakage issue during the link prediction. We randomly partition the triples without reverse relations by the ratio of 7:3 to obtain the training set and testing set. After that, the corresponding triples with reverse relations are added to these sets.

We set the hyperparameters of our model as follows. The batch size is in accordance with the size of the training set. Both the vector dimension in the structural and semantic embeddings is 100, and thus the initialization vector dimension is 200. The weight attenuation coefficient of the GAT is $1e^{-5}$. The learning rate is 0.001. We set the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 1e^{-8}$.

We compare our model with the four recent embedding-based methods in order to verify the effectiveness of the representation learning for the entities and relations in the ICC-Vulnerability KG. We further select the entity prediction and calculate three metrics by Eq. (8). The link prediction results are demonstrated in Table 4. It is clearly observed that the model in this work outperforms the other four methods. The MR is lowest, while the $Hits@k$ ($k = 1, 3, 10$) and MRR are highest. There are many entities with 1 to N relations in our KG. Hence, embeddings in conventional methods fail to predict most of the entities. It also indicates that our model captures their features by aggregating the adjacent entities and relations, which guarantees the representation quality of path vectors in the subsequent parts.

Table 4: Link prediction results

	MR	MRR	Hits@1	Hits@3	Hits@10
TransE [27]	3319	0.102	0.068	0.116	0.166
TransH [30]	3276	0.110	0.090	0.113	0.149
ConvKB [31]	4657	0.070	0.057	0.072	0.093
R-GCN [32]	3737	0.097	0.085	0.100	0.119
Model without structural and semantic Embeddings	1076	0.225	0.189	0.234	0.294
Our model	866	0.264	0.225	0.274	0.336

We carry out an ablation experiment where the model in our work removes the semantic and structural embeddings. The metrics of the link prediction are also listed in Table 4. Fig. 7 shows the trend of each metric as epochs increase, and it is inferior to that in our model. Fig. 8 is the training

loss comparison. Our model has faster convergence. It is obvious that the initialization vectors can not only obtain a good representation for the KG but also speed up the training.

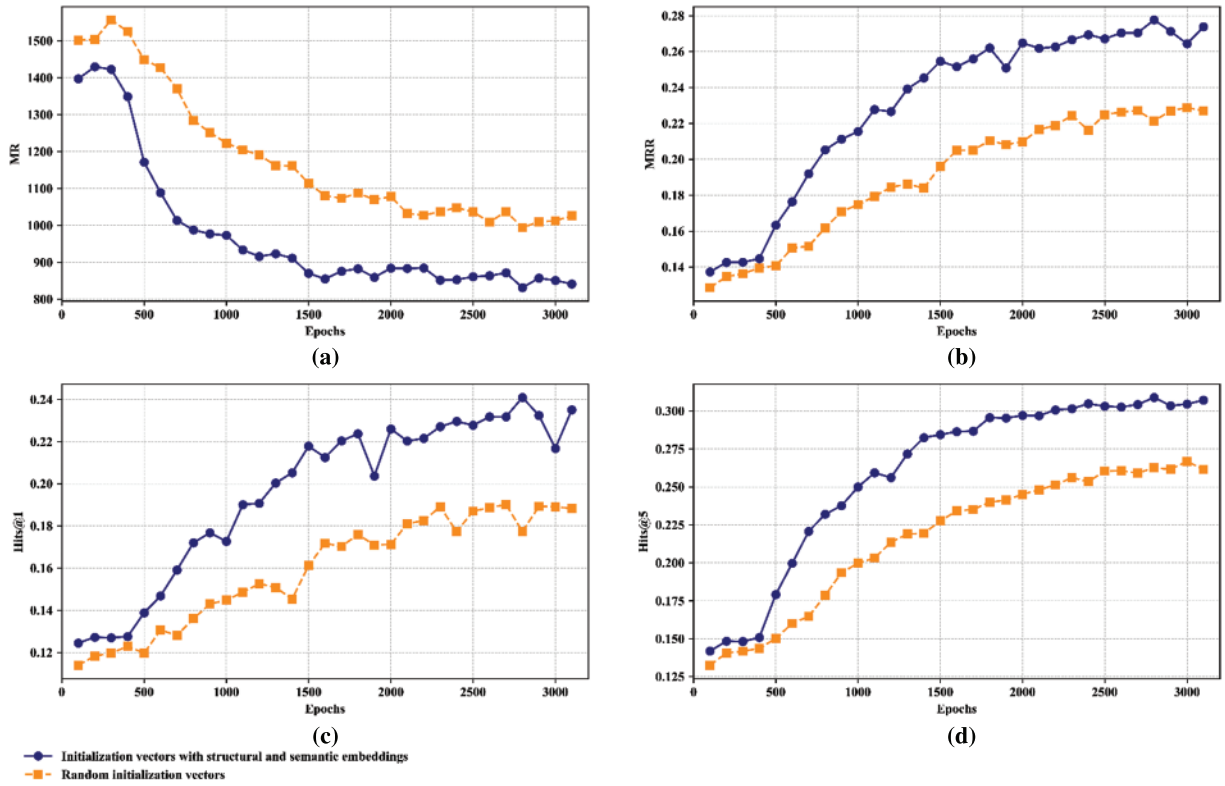

Figure 7: Link prediction metric evaluations with respect to the initialization vectors with structural and semantic embeddings

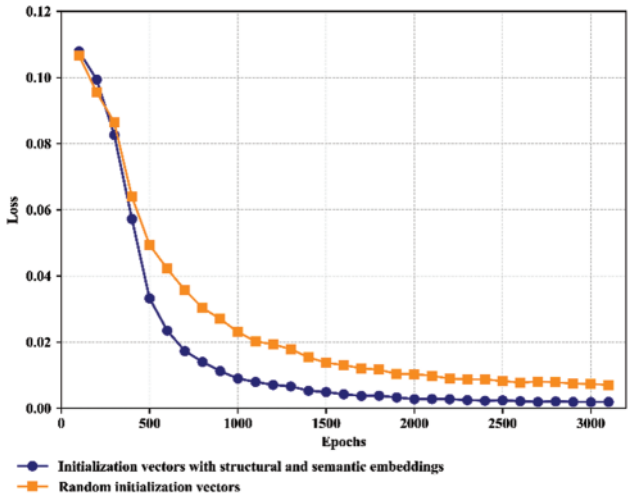

Figure 8: Training loss with respect to the initialization vectors with structural and semantic embeddings

7.3 Relation Reasoning Results

- Data Processing

Based on the KG embeddings in the previous subsection, we process the data and make it suitable for path-based reasoning tasks. In this part, the query relation is the *can_exp* relation. A new dataset consists of all entity pairs of the Attacker entities and the *E-Vul.* entities that are recorded as multiple pairs (*Attacker*, *E-Vul.*), and relation paths between each pair that are stored in the vector form. If a selected entity pair is connected by the *can_exp* relation, it will be assigned to the positive sample set; otherwise, it will belong to the negative sample set. There are two reasons why we only focus on the pairs (*Attacker*, *E-Vul.*) in this part. In the practical aspect, the security assessment objective is often explicit and all potential evidence for each specific exploitable vulnerability has been captured by the relation paths. It is unnecessary for analysts to analyze other entity pairs. In a data aspect, the *can_exp* relation only has a correlation with the pairs (*Attacker*, *E-Vul.*). The representation of the relation path also omits that of the passing entity and only considers the collection of the relevant relations. We still randomly partition the dataset by the ratio of 7:3 to obtain the training and testing sets as illustrated in Table 5.

- Critical Relation Path Selection

We compare the functional results and performance of the relation path search algorithm with and without the similarity calculation. In the first step, we randomly select a pair (*Attacker*, *E-Vul.*). Two search algorithms both set the maximum path length to 4 and the discovered relation paths are listed in Table 6. The first two columns of the table are the results of the proposed method, and the others are the search method without the similarity calculation. Each relation path is represented by the sequence of relation indexes. Some of the relation indexes can be found in Table 3, and the index plus 12 is recorded as the index of the corresponding reverse relation. Meanwhile, the number of the same relation paths selected by these two algorithms is recorded. By comparing the results, it is observed that all critical relation paths between the pairs of entities can be selected. The same relation paths that pass through different entities are redundant for subsequent reasoning tasks and the path can be removed by the proposed algorithm.

Table 5: Dataset statistics of the relation path reasoning

	Test	Train	Total
Positive samples	1568	686	2254
Negative samples	1243	520	1763
Total	2811	1206	4017

Table 6: Relation paths between the selected pair (*Attacker*, *E-Vul.*)

Proposed algorithm		Relation path search algorithm without the similarity calculation					
Relation path	Path num.	Relation path	Path num.	Relation path	Path num.	Relation path	Path num.
1, 18, 11	1	1, 18, 17, 11	14	1, 24	1	3, 21, 8, 20	9

(Continued)

Table 6: Continued

Proposed algorithm		Relation path search algorithm without the similarity calculation					
Relation path	Path num.	Relation path	Path num.	Relation path	Path num.	Relation path	Path num.
1, 18, 10, 20	1	1, 24, 8, 20	11	1, 24, 23, 11	2	3, 21	1
1, 18, 10, 21	1	1, 24, 9, 21	25	1, 24, 4, 16	45	3, 21, 23, 11	2
1, 24	1	1, 18, 11	1	3, 21, 4, 16	45	3, 21, 12, 24	1
3, 21	1	1, 18, 10, 20	1	1, 18, 10, 21	1	3, 22, 17, 11	22

Next, we evaluate the impacts of the maximum path length on the performance of the respective search processes. Ten pairs (*Attacker*, *E-Vul.*) are randomly selected as candidates for the performance evaluation. With the path length ranging from 2 to 6, the trend of the running time and memory usage of the above relation path search algorithms is illustrated in Fig. 9.

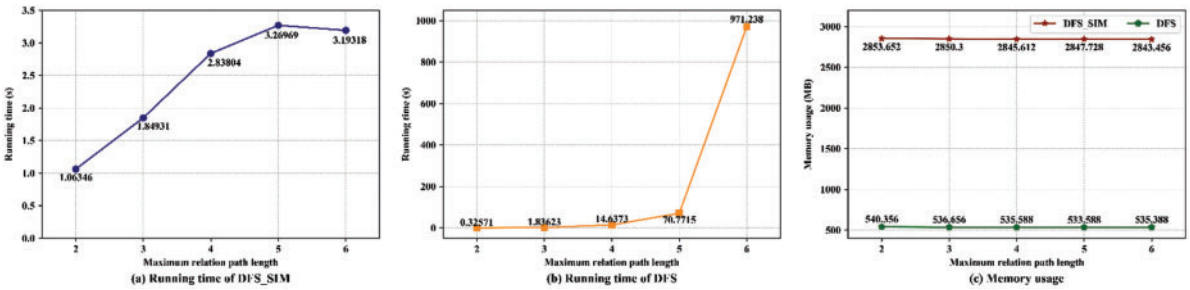

Figure 9: Performance evaluation of the critical relation path depth-first search with similarity calculation

More relation paths are discovered when the maximum path length is set to 2 or 3, so the proposed algorithm needs more running time spent on the similarity calculation than the other search methods. As the path length increases, the advantage of the proposed method appears in the running time, and it takes about 3 s to discover all the paths. In contrast, the running time sharply increases in the search algorithm without the similarity calculation. Moreover, the memory usage of the proposed algorithm is nearly six times as large as that of the compared algorithm. It is reasonable for the proposed algorithm that the similarity calculation needs temporary storage for a number of relation vectors. Fortunately, memory usage tends to be stable with increasing path length.

- Aggregation-Based Reasoning

After the acquirement of a collective of critical relation paths, we will aggregate them for the accomplishment of the query relation reasoning. The hyperparameters of the critical relation path-based model are given as follows. The dimension of the projection matrix in the attention layer is set to 400. The weight attenuation coefficient is $1e^{-6}$. The learning rate is 0.005. The other parameters are the same as those in Subsection 7.2.

Loss and accuracy curves of the training and testing sets during the 800 epochs are respectively illustrated in Fig. 10. It is observed that the loss curves of the training and test sets converge at almost

the same time. In addition, the proposed method has high accuracy. The accuracy of the training and test sets has almost the same trend, so there is no over-fitting during the training process.

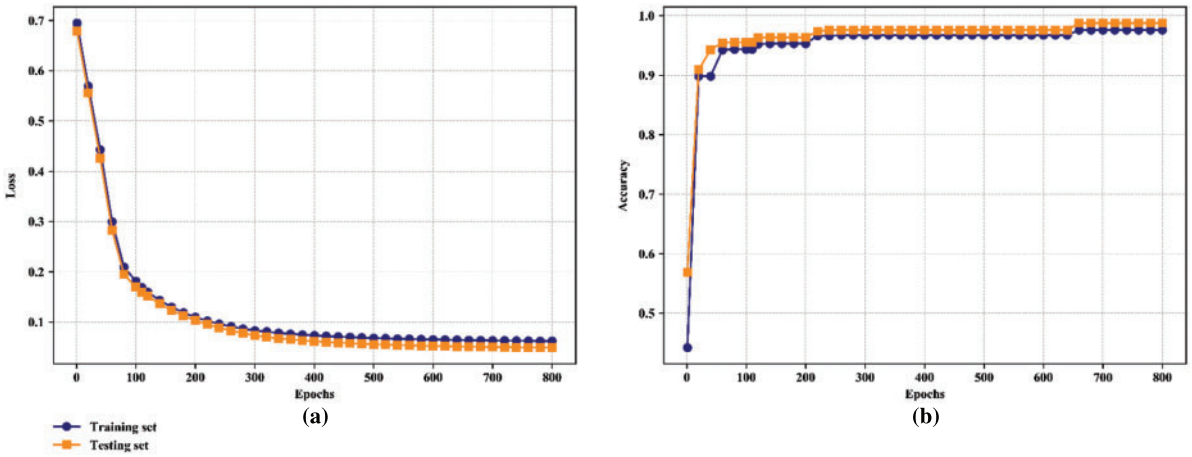

Figure 10: Training loss and accuracy of the training set and the test set

Leveraging the well-trained model, we obtain seven critical relation paths and their attentive weights, and they are listed in Table 7. Among the discovered paths, four are assigned higher weights than the remaining three. Take two of them as an example to demonstrate the practical interpretability of the model. The first path has the highest attentive weight since the relation path indicates that the exploitable vulnerability can be remotely accessed via a network and the potential attacker has an opportunity to exploit it. The last path in the table has the least attentive weight since it merely implies the effect on the vulnerability once the privilege is gained after an exploit, but it is uncertain whether the attacker has the privilege for the current component.

Table 7: Attentive weights of each critical path

No.	Relation path	Weight
1	assess, net_vector^{-1}	0.559
2	assess, conn^{-1}, has_vul	0.1676
3	gain_pri, has_pri^{-1}	0.1134
4	assess, conn^{-1}, has_pri, pre_con^{-1}	0.1021
5	assess, conn^{-1}, has_pri, pos_con	0.0312
6	assess, conn^{-1}, has_app, has_vul	0.0173
7	gain_pri, pos_con^{-1}	0.0090

Aiming to validate the effectiveness of reasoning in the KG, we choose two path-based methods that are widespread in similar work, namely the PRA [7] and the PTransE [6]. As mentioned in the related work, the PRA employs a random walk algorithm for the extraction of the individual path feature and a supervised classification for the determination of the query relation. A logistic regression classifier is built in this part for the reasoning given by the PRA. Introducing the path feature into embeddings, the PTransE implements a score function to deal with the vectors of all selected entity

pairs and the query relation. A linear classifier utilizes the scoring results to decide whether the query relation is valid.

Furthermore, we carry out two ablation experiments. (1) The attention layer in the model is removed and the aggregative vector with an average vector of the critical paths between an entity pair is added instead. (2) The proposed method removes the similarity calculation just like the methods mentioned in [8,22], which means the redundant relation paths are left. Experimental results of those four methods and the proposed one are shown in Table 8. Our model has superior results in terms of precision, F1 score, and accuracy. Simultaneously, it is essential to aggregate critical paths by means of the attention layers for improving classification performance. It is proved that the proposed method has the advantage of query relation reasoning over the other three methods.

Table 8: Experimental results of reasoning

	Precision	Recall	F1 score	Accuracy
PTansE [6]	0.9123	0.8790	0.8953	0.8831
PRA [7]	0.9444	0.8921	0.9175	0.9087
Proposed method without attention layers	0.9740	0.9840	0.9790	0.9760
Proposed method without similarity calculation [8,22]	0.9687	0.9927	0.9806	0.9776
Proposed method	0.9941	0.9840	0.9890	0.9876

8 Conclusion

In this work, we propose a critical relation path aggregation-based KG reasoning approach for the identification of the ICC exploitable vulnerabilities. The approach consists of ICC-Vulnerability KG construction, relation path representation, as well as query relation reasoning. The KG is driven by a security ontological model in the view of the potential attacker to integrate the vulnerability knowledge and scene context from multiple sources. We focus on the relation paths that contain a wealth of evidence to support the subsequent exploitability analysis. Depending on the KG embedding technique, the relation path representation is achieved by accumulating all relation vectors on the path. And each vector is obtained after aggregating the embeddings with the introduction of the local structure and related semantics for improvements in representation performance. Next, a critical relation path selection algorithm is developed based on the similarity calculation between each path vector and the query relation vector. It is inspired by similarity increment and timely removal of redundant relation paths that have no help for reasoning. Ultimately, an aggregative path vector with favorable evidence is generated by using the collective critical paths and a binary classifier is built for the determination of the query relation validity to accomplish the reasoning tasks. Note that these two aggregations are guaranteed by the attention mechanism. Experimental results show that the proposed method performs better than the state-of-the-art ones in the aspects of embedding quality and query relation reasoning accuracy.

In the future, the extraction of entities and relations will be improved with the ongoing natural language processing techniques, instead of the rule-based extraction in the construction of the KG. Additionally, multi-hop reasoning is a research direction for the connection discovery among different relations. To achieve that goal, integrating the convolutional neural network-based encoding into the relation path representation is a promising way to learn more features of entities and relations.

Funding Statement: Our work is supported by the National Key R&D Program of China (2021YFB2012400).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. Upadhyay and S. Sampalli, "SCADA (Supervisory control and data acquisition) systems: Vulnerability assessment and security recommendations," *Computers & Security*, vol. 89, pp. 101666–101683, 2020.

[2] S. D. D. Anton, D. Fraunholz, D. Krohmer, D. Reti, D. Schneider *et al.,* "The global state of security in industrial control systems: An empirical analysis of vulnerabilities around the world," *IEEE Internet Things Journal*, vol. 8, no. 24, pp. 17525–17540, 2021.

[3] G. Yadav, P. Gauravaram, A. K. Jindal and K. Paul, "SmartPatch: A patch prioritization framework," *Computers in Industry*, vol. 137, pp. 103595–103608, 2022.

[4] E. Kiesling, A. Ekelhart, K. Kurniawan and F. Ekaputra, "The SEPSES knowledge graph: An integrated resource for cybersecurity," in *Proc. ISWC*, Auckland, New Zealand, pp. 198–214, 2019.

[5] M. Eckhart, A. Ekelhart and E. Weippl, "Automated security risk identification using AutomationML-based engineering data," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1655–1672, 2022.

[6] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao *et al.,* "Modeling relation paths for representation learning of knowledge bases," in *Proc. EMNLP*, Lisbon, Portugal, pp. 705–714, 2015.

[7] Y. Jia, Y. Qi, H. Shang, R. Jiang and A. Li, "A practical approach to constructing a knowledge graph for cybersecurity," *Engineering*, vol. 4, no. 1, pp. 53–60, 2018.

[8] X. Jiang, Q. Wang, B. Qi, Y. Qiu, P. Li *et al.,* "Attentive path combination for knowledge graph completion," in *Proc. ACML*, Seoul, Korea, pp. 590–605, 2017.

[9] B. Jagvaral, W. K. Lee, J. S. Roh, M. S. Kim and Y. T. Park, "Path-based reasoning approach for knowledge graph completion using CNN-BiLSTM with attention mechanism," *Expert Systems with Applications*, vol. 142, pp. 112960–112969, 2020.

[10] S. Wu, Y. Zhang and W. Cao, "Network security assessment using a semantic reasoning and graph based approach," *Computers and Electrical Engineering*, vol. 64, pp. 96–109, 2017.

[11] Z. Wang, H. Zhu, P. Liu and L. Sun, "Social engineering in cybersecurity: A domain ontology and knowledge graph application examples," *Cybersecurity*, vol. 4, no. 1, pp. 1–21, 2021.

[12] J. Alanen, J. Linnosmaa, T. Malm, N. Papakonstantinou, T. Ahonen *et al.,* "Hybrid ontology for safety, security, and dependability risk assessments and security threat analysis (STA) method for industrial control systems," *Reliability Engineering and System Safety*, vol. 220, pp. 108270–108289, 2022.

[13] R. Syed, "Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system," *Information & Management*, vol. 57, no. 6, pp. 103334–103350, 2020.

[14] D. Du, X. Ren, Y. Wu, J. Chen, W. Ye *et al.,* "Refining traceability links between vulnerability and software component in a vulnerability knowledge graph," in *Proc. ICWE*, Cáceres, Spain, pp. 33–49, 2018.

[15] I. Sarhan and M. Spruit, "Open-CyKG: An open cyber threat intelligence knowledge graph," *Knowledge-Based Systems*, vol. 233, pp. 107524–107536, 2021.

[16] G. Shen, W. Wang, Q. Mu, Y. Pu, Y. Qin *et al.,* "Data-driven cybersecurity knowledge graph construction for industrial control system security," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–13, 2020.

[17] Y. Wang, Y. Zhou, X. Zou, Q. Miao and W. Wang, "The analysis method of security vulnerability based on the knowledge graph," in *Proc. ICCNS*, Tokyo, Japan, pp. 135–145, 2020.

[18] S. Zhang, G. Bai, H. Li, P. Liu, M. Zhang *et al.,* "Multi-source knowledge reasoning for data-driven IoT security," *Sensors*, vol. 21, no. 22, pp. 7579–7597, 2021.

[19] S. Qin and K. P. Chow, "Automatic analysis and reasoning based on vulnerability knowledge graph," in *Proc. CCIS*, Beijing, China, pp. 3–19, 2019.
[20] Z. Han, X. Li, H. Liu, Z. Xing and Z. Feng, "DeepWeak: Reasoning common software weaknesses via knowledge graph embedding," in *Proc. SANER*, Campobasso, Italy, pp. 456–466, 2018.
[21] H. Xiao, Z. Xing, X. Li and H. Guo, "Embedding and predicting software security entity relationships: A knowledge graph based approach," in *Proc. ICNIP*, Sydney, NSW, Australia, pp. 50–63, 2019.
[22] L. Yuan, Y. Bai, Z. Xing, S. Chen, X. Li *et al.,* "Predicting entity relations across different security databases by using graph attention network," in *Proc. COMPSAC*, Madrid, Spain, pp. 834–843, 2021.
[23] N. Lao, T. Mitchell and W. W. Cohen, "Random walk inference and learning in a large scale knowledge base," in *Proc. EMNLP*, Edinburgh, Scotland, UK, pp. 529–539, 2011.
[24] G. Niu, B. Li, Y. Zhang, Y. Sheng, C. Shi *et al.,* "Joint semantics and data-driven path representation for knowledge graph reasoning," *Neurocomputing*, vol. 483, pp. 249–261, 2022.
[25] J. Huang, T. H. Zhang, J. Zhu, W. Yu, Y. Tang *et al.,* "A deep embedding model for knowledge graph completion based on attention mechanism," *Neural Computing and Applications*, vol. 33, no. 15, pp. 9751–9760, 2021.
[26] M. U. Aksu, K. Bicakci, M. H. Dilek, A. M. Ozbayoglu and E. İ. Tatlı, "Automated generation of attack graphs using NVD," in *Proc. CODASPY*, Tempe, AZ, USA, pp. 135–142, 2018.
[27] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. NIPS*, Lake Tahoe, NV, USA, vol. 26, pp. 1–9, 2013.
[28] D. Nathani, J. Chauhan, C. Sharma and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in *Proc. ACL*, Florence, Italy, pp. 4710–4723, 2020.
[29] T. Sasaki, A. Fujita, C. H. Ganan, M. van Eeten, K. Yoshioka *et al.,* "Exposed infrastructures: Discovery, attacks and remediation of insecure ICS remote management devices," in *Proc. IEEE S&P*, San Francisco, CA, USA, pp. 2379–2396, 2022.
[30] Z. Wang, J. Zhang, J. Feng and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI*, Québec, Canada, pp. 1112–1119, 2014.
[31] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proc. NAACL HLT*, New Orleans, LA, USA, pp. 327–333, 2018.
[32] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov *et al.,* "Modeling relational data with graph convolutional networks," in *Proc. ESWC*, Heraklion, Crete, Greece, pp. 593–607, 2018.