sensors

MDPI

Article

Multi-Source Knowledge Reasoning for Data-Driven IoT Security

Shuqin Zhang [1,*], Guangyao Bai [1], Hong Li [2], Peipei Liu [2], Minzhi Zhang [1] and Shujun Li [3]

[1] School of Computer Science, Zhongyuan University of Technology, Zhengzhou 450007, China; 2019107118@zut.edu.cn (G.B.); 2019107117@zut.edu.cn (M.Z.)
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China; lihong@iie.ac.cn (H.L.); liupeipei@iie.ac.cn (P.L.)
[3] School of Information Science and Technology, Yancheng Teachers University, Yancheng 224002, China; lisj@yctu.edu.cn
* Correspondence: zhangsq@zut.edu.cn

Abstract: Nowadays, there are different kinds of public knowledge bases for cyber security vulnerability and threat intelligence which can be used for IoT security threat analysis. However, the heterogeneity of these knowledge bases and the complexity of the IoT environments make network security situation awareness and threat assessment difficult. In this paper, we integrate vulnerabilities, weaknesses, affected platforms, tactics, attack techniques, and attack patterns into a coherent set of links. In addition, we propose an IoT security ontology model, namely, the IoT Security Threat Ontology (IoTSTO), to describe the elements of IoT security threats and design inference rules for threat analysis. This IoTSTO expands the current knowledge domain of cyber security ontology modeling. In the IoTSTO model, the proposed multi-source knowledge reasoning method can perform the following tasks: assess the threats of the IoT environment, automatically infer mitigations, and separate IoT nodes that are subject to specific threats. The method above provides support to security managers in their deployment of security solutions. This paper completes the association of current public knowledge bases for IoT security and solves the semantic heterogeneity of multi-source knowledge. In this paper, we reveal the scope of public knowledge bases and their interrelationships through the multi-source knowledge reasoning method for IoT security. In conclusion, the paper provides a unified, extensible, and reusable method for IoT security analysis and decision making.

Keywords: IoT security; threat analysis; ontology; knowledge reasoning; inference rules

check for updates

Citation: Zhang, S.; Bai, G.; Li, H.; Liu, P.; Zhang, M.; Li, S. Multi-Source Knowledge Reasoning for Data-Driven IoT Security. *Sensors* **2021**, *21*, 7579. https://doi.org/10.3390/s21227579

Academic Editor: Ilsun You

Received: 30 August 2021
Accepted: 11 November 2021
Published: 15 November 2021

1. Introduction

With the development of the Internet of Things (IoT), the massive deployment of IoT devices and the popularization of IoT technology have facilitated people's lives [1]. As many organizations transform their devices to adopt IoT-connected technologies, it is estimated that there will be more than 40 billion IoT devices by 2027, with the potential value of the IoT going up to USD 11 trillion by 2025 [2]. However, the hidden security issues of the IoT are gradually emerging. Recent years have witnessed some of the largest, most sophisticated, and most severe cyber attacks, such as the SolarWinds attack [3] and the NetSarang malicious code attack [4], which affected millions of consumers and thousands of businesses. The in-depth integration of informatization and industrialization, such as the industrial control network of the mining, electric, and chemical industries, or the internal network of government, military, finance, etc., have all gradually developed from mutually independent and closed networks to interconnected and open ones [5]. The traditional boundary between the internal and the external networks is gradually being blurred, which consequently introduces ubiquitous network security risks. The integration of industrial control devices and network technology into the IoT environment eliminates the internal and external security boundaries of the industrial environment. Industrial control network devices do not only have security risks such as those affecting the security of software and hardware on the transmission link and the blurring of network protection boundaries, but they are also exposed to new threats such as a stepping attack, data sniffing, blockage, and eavesdropping within the IoT Cloud platform service. Hence, IoT security is challenging. Attacks on IoT devices are becoming more intelligent and diversified. Meanwhile, with massive numbers of IoT devices accessing the Internet, the interaction of heterogeneous information and rapid changes in the network structure have further expanded the scope of attacks and have led the IoT environment to continuously generate new weaknesses and threats [6]. When new threats are encountered, traditional security intrusion detection and response technologies cannot adapt to the attacks faced by IoT security. Security Information and Event Management (SIEM) and Security Operation Center (SOC) [7] also have limitations. Their mechanisms are often built on heuristic and static attack signatures and cannot detect new variants of attacks [8]. However, IoT systems are vulnerable to new families of attacks that could exploit the attack surfaces of devices and their network protocols. The SIEM system has a lack of security considerations for protecting their heterogeneous and complex devices and systems. IoT security requires an effective method to intelligently respond to security intrusions.

A variety of intelligent reasoning technologies can be used to recognize threats to the IoT. For example, the reasoning technology and semantic-web technology based on ontology [9], and the text mining and malicious code detection technology based on Natural Language Processing (NLP) [10,11]. However, due to the heterogeneity and complexity of the IoT, it is very difficult to create procedures for global security status detection and threat awareness in the IoT system. There are still challenges to security management and threat analysis within the IoT system.

This paper proposes a multi-source knowledge reasoning method for IoT security. We integrate knowledge to solve the semantic heterogeneity of multi-source knowledge by analyzing the characteristics of the security knowledge base for the IoT. In addition, we model an ontology to describe the elements of IoT security threats and design inference rules for threat analysis. This method can perceive the security status within the IoT environment and automatically infers mitigations to improve the threat response capability of the system.

Our contributions in this paper are as follows:

- We extracted the relationship between the entries of the IoT security public knowledge bases for knowledge integration, and the relationship mapping link graph model is constructed to provide support for the assessment of threat elements that affect IoT security;
- An IoT security threat ontology framework is proposed to describe the correlation of threat objects. The framework expands the current knowledge domain of network security ontology modeling and can provide a wider sense of security status;
- This paper proposes a reasoning method based on the multi-source knowledge of IoT security, which can perceive highly vulnerable platforms in the IoT environment and automatically respond to threats.

The remainder of the study is organized as follows: Section 2 discusses the literature review. Section 3 presents in-depth information on the IoT security multi-source knowledge base. Section 4 presents information on the methodology used in developing the proposed framework. After this, we formalize the classes of the proposed ontology model. Section 5 presents several scenarios to further illustrate the feasibility and effectiveness of the proposed model. Finally, we conclude the whole paper in Section 6.

2. Related Work

At present, there has been a lot of basic research around vulnerabilities and latent threats. Network security experts can formulate defense strategies, skills, and operations by using public structured description language and public cyber security knowledge bases. The literature analysis conducted suggests that there are several initiatives providing support for IoT threat analysis, predominantly through the use of threat modeling [12–14], knowledge graph [15], and graph theory [16] technologies. Syed et al. [12] integrated heterogeneous data, knowledge models, and common network security standards for information sharing and exchange from different network security systems. They built a unified cyber security model called Unified Cybersecurity Ontology (UCO), which supports information integration and network situation awareness. Abbas et al. [13] applied the STRIDE threat modeling method to the smart autonomous vehicular system and smart home, which identifies and mitigates threats that may lead to phishing attacks. Huang et al. [14] employed the anonymity of blockchain to protect identity information in threat intelligence. This work used encrypted threat intelligence to construct a complete attack chain and used a blockchain-based threat intelligence sharing framework for correlation analysis. Kiesling et al. [15] designed an evolving cyber security knowledge graph by integrating and linking critical information from multiple public knowledge bases such as vulnerabilities, weaknesses, and attack patterns; they also provided use cases for vulnerability assessment and sample queries for intrusion detection. Tian et al. [16] used the graph theory to model the network structure based on the complex characteristics of the network structure on the Internet. They also calculated the security situation of network nodes based on threat propagation, which can quantify the boundary link relationship of security protection.

However, the current research has not formally described the relationship between the core concepts of IoT security, which are unable to define inference rules because the research on threat analysis has not yet been improved to the semantic level. In this paper, we integrate the heterogeneous, multi-source knowledge bases for cyber security and uniformly represent the data in a graph; this graph will provide support for the next work on knowledge inference through the context of semantic information from different knowledge bases.

Ontology is used to describe information objects and support the sharing and reusing of domain knowledge. In Computer Science, an ontology is a formal description of concepts and relationships for an application domain of the real entity. A central aspect is the sharing of knowledge and information with the use of a common vocabulary, as supported by the Resource Description Framework (RDF). Ontology modeling is a means to formally model the structure of a system, which is useful for Cyber Threat Intelligence (CTI) analysis. At present, researchers have developed the open ontology for the security of IoT ecosystem elements [17–20], relating them with existing security concepts, primitives, weaknesses, vulnerabilities, and practices [21–24]. Researchers have conducted in-depth research on CTI analysis and ontology modeling in cyber security. Jia et al. [17] built a cyber security ontology based on assets, vulnerabilities, and attacks; they proposed a practical method for constructing a cyber security knowledge graph, and inferred new rules based on the five-tuple model of the cyber security knowledge base. Rastogi et al. [18] designed a malware ontology called MALOnt, which contained concepts such as malware characteristics, attack behavior, and detailed information about the attacker. It supported the collection of intelligence on malware threats from different online sources, and built a knowledge graph framework based on MALOnt. Mozzaquatro et al. [23] proposed an IoT network security framework for knowledge reasoning. It integrated knowledge on known network security issues (e.g., vulnerabilities and known threats) and the corresponding preventive measures into IoTSec [19] ontology. It enabled the security system to automatically detect threats to the IoT and to dynamically propose or implement the appropriate protection services. Choi et al. [20] proposed a security service framework that could be used in the power IoT-Cloud environment by analyzing the security vulnerabilities of the power system in that environment and by modeling the security context ontology. The framework used smart meters as an example to create an attack scenario in the power IoT, and realized a security mechanism that could operate effectively in this environment. Si et al. [21] proposed a knowledge base model of network security situation elements. The model was based on elements such as vulnerability attributes (e.g., severity of vulnerability, access request, result type, and distribution), vulnerability objects (e.g., configuration errors, protocol vulnerabilities), and constructed the ontology of the network security vulnerability. Li et al. [22] proposed a general network security parameter classification architecture, which extended the vulnerability ontology in the host domain based on the type of inheritance relationship.

Nevertheless, most of the current ontology modeling of cyber security are focused on vulnerabilities, weaknesses, and attack patterns. The research that combines the attacker's Tactics, Techniques, and Procedures (TTP) with vulnerabilities and weaknesses is still in the exploratory stage. Therefore, it is difficult to perform multi-step attack prediction, threat analysis, and the follow-up work. The ontology model proposed in this paper expands the current knowledge domain of cyber security ontology modeling. This model provides a broader awareness of security status and improves threat response capabilities.

3. IoT Security Multi-Source Knowledge Base

Massive amounts of security information are fragmented and scattered on the Internet. The public information security knowledge bases maintained by the security organizations MITRE and NIST have gradually been developed into well-known public industry standards in cyber security. IoT security research results can be shown and shared through a set of structured information security description standards and specifications. We data mined a set of these information sources in order to expand upon their defensive utility in threat analysis. Well-known public knowledge bases in cyber security that we used in our work include:

- Common Vulnerabilities and Exposures (CVE) [25];
- National Vulnerability Database (NVD) [26];
- Common Weakness Enumeration (CWE) [27];
- Common Attack Pattern Enumeration and Classification (CAPEC) [28];
- Common Platform Enumeration (CPE) [29];
- Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) matrix [30].

3.1. Data Sources

CVE discloses the exposed vulnerabilities. A vulnerability is defined as a "weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact on confidentiality, integrity, or availability" [26]. Each vulnerability in the database has an identification number and a related description defined by MITRE. NVD provides information about security-related software vulnerabilities, product configurations, and impact indicators. NVD is built on the CVE list and is fully synchronized with it as well. NVD provides enhanced information for entries in the CVE list, such as structured information, severity score, and impact level given by the Common Vulnerability Scoring System (CVSS) [31]. CVSS is an industry public standard for evaluating the severity of vulnerabilities. Most vulnerability severity research and commercial vulnerability management platforms are evaluated based on CVSS. CAPEC provides a summary of the attack pattern classification and focuses on the attacker's use of cyberspace vulnerabilities. Security managers who understand attack patterns are essential to threat analysis and defense. CPE entries are specifically of interest because cyber security managers can scan them in order to be alerted to specific targets in their IoT systems. "Weaknesses are flaws, faults, bugs, and other errors in software and hardware design, architecture, code, or implementation that, if left unaddressed, could result in systems, networks, and hardware being vulnerable to attack" [32]. Information on the weaknesses is summarized by the CWE.

3.2. ATT&CK Matrix

The ATT&CK matrix was first proposed by MITRE in 2013. Through the summarization and analysis of real observational data and Advanced Persistent Threats (APT) [33], ATT&CK has gradually developed into a general language for attackers' behavior description and a behavior analysis model for the entire life cycle of the attack chain. ATT&CK abstractly describes a framework composed of sequential network attack tactics, each of which covers abundant attack techniques. From the perspective of attack detection and threat analysis, the context information associated with the attack can be further speculated only when the attack tactics and techniques are clearly defined. ATT&CK persistently constructs and enriches the attackers' tactics and techniques in order to help researchers to master the global attack technique needed to support the assessment and automatically respond to security intrusions.

Researchers are currently focusing on the feasibility of applying the ATT&CK matrix to threat analysis. In security intelligence research, vulnerability intelligence, which is mostly from the perspectives of software, hardware, operating system, and protocol weaknesses, developed earlier and is more mature [34]. However, threat intelligence mainly collects external factors related to attackers or attack behaviors [35]. Security managers achieve timely management and control of threats by integrating threat information and facilitating the sharing threat information [36]. Apart from the inherent complexity of the IoT, the heterogeneous information exchange between IoT devices and systems further aggravates its structural complexity [37]. Researching on vulnerability intelligence has great limitations in the complex environment of the IoT. Therefore, researchers hope to conduct analyses of threats in the IoT environment by analyzing and understanding the attackers' targets and systemic risks. The ATT&CK matrix can connect threat events and observation data, and can further open up the link to promote an understanding of the threats [38].

3.3. Knowledge Integration and Relationship Mapping

It is worth noting that Attack Patterns in CAPEC connect the ATT&CK matrix to the CWE source, functioning as bridges that link a Technique within a Tactic to a CWE entry. Meanwhile, a CWE entry has a relational link to a CVE entry. The relationship implies that the Vulnerability is an example of the Weakness. In this paper, we use the Attack Patterns as bridges that relate a means of attack, i.e., Tactic and Technique, to its targeted Weakness. A Weakness in CWE can be linked to a Vulnerability in CVE, and it can be linked to a known affected Platform in CPE. Security researchers can select a particular application, hardware, or operating system in cyberspace to see which Tactics or Techniques will be affected by this end-to-end linkage method. Figure 1 illustrates the relational linkage mapping based on the selected knowledge bases.

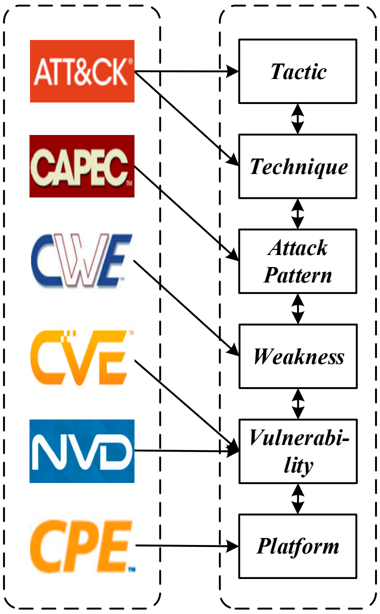


Figure 1. Schematic of relational mapping linkages.

Based on the mapped relationship links, this paper combines the source data and structural characteristics of each knowledge base, and uses a graph structure to uniformly represent the data. Each layer represents a different source. The nodes of the graph represent the entries of the knowledge bases. The internal and external links between the knowledge bases are retained and represented by the edges of the graph. These edges are not bi-directional in the source knowledge bases. However, when integrated into the unified graph structure data, it can be traversed bi-directionally and it is easy to trace the relationship between different knowledge sources from any node. In a nutshell, ATT&CK provides the Tactics and Techniques used by attackers on vulnerable systems. CWE, CVE, and CPE reveal the positions of Weaknesses and Vulnerabilities exploited by attackers in the IoT. Moreover, CAPEC associates potential attacks with Weaknesses that may become targets.

4. Ontology-Based Multi-Source Knowledge Reasoning Framework for IoT Security

Attacks on heterogeneous networks are the most prominent in the IoT environment [39]. The proposed method in the aforementioned section provides links to the understanding of threats and an overall view of the IoT security status. However, awareness of the network situation in multi-source heterogeneous IoT environments is a challenge. Ontology plays an important role in solving the semantic heterogeneity of CTI through the formal description of specific domain knowledge [40,41]. We use the Web Ontology Language (OWL) [42] to build a unified formal description. Concepts are implemented as classes, and relationships are implemented as properties. The expressive ability of OWL is limited to description logic and cannot express uncertain knowledge such as the changes of events in spatio-temporal and semantic relations. In order to enhance the reasoning ability of this model, the second half of this paper uses the semantic web rule language to design inference rules that complement the description ability of ontology. This paper proposes an IoT security knowledge reasoning model based on semantic ontology and rule logic. Our vision is to present a novel approach that improves IoT cybersecurity awareness of situation and focuses on the fusion of multi-source heterogeneous knowledge and the analysis of vulnerabilities, weaknesses, attack patterns, techniques, and tactics in a unified knowledge base. This approach will also enable the subsequent security service provisioning adjusted to the dynamic threat intelligence analysis, hence improving the security response mechanisms around threat intrusion and IoT assets.

Figure 2 shows the multi-source knowledge reasoning framework for IoT security. The framework consists of a data and ontology repository and a reasoning engine. The heterogeneous data of multiple knowledge bases are preprocessed into unified graph data. The ontology model is constructed based on the integrated knowledge base characteristics and the external ontology model. In the inference layer, there are inference rules designed according to a specific IoT environment and applied to threat response modules. The workflow of the proposed framework is as follows:

1. The multi-source heterogeneous IoT security knowledge is obtained from crawlers embedded in several knowledge sources. The amount of knowledge is huge and the structure of the knowledge is different;
2. The crawled multi-source heterogeneous knowledge is integrated into a unified graph database;
3. The integrated data are mapped into the proposed ontology model through instance mapping, and the generated instances are integrated into the ontology repository;
4. The inference engine perceives and separates the abnormalities based on the instances repository and the user-defined inference rules to achieve the goal of automatically responding to threats.

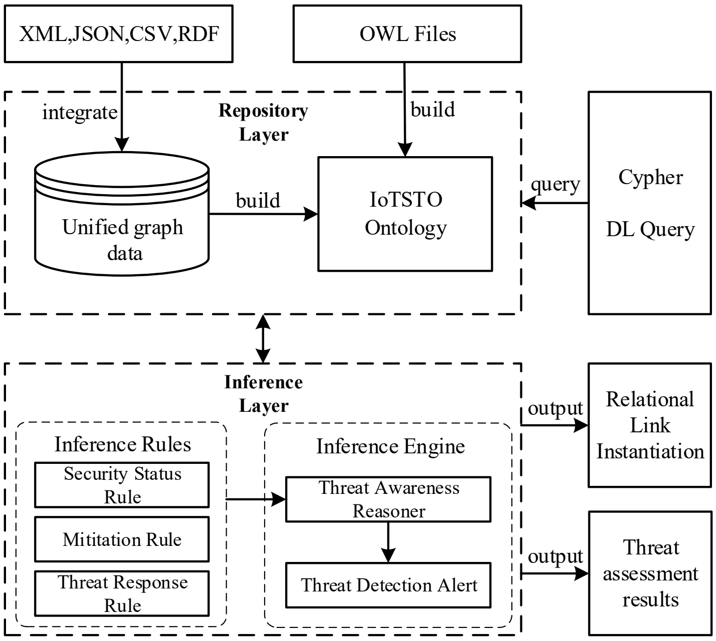


Figure 2. Multi-source knowledge reasoning framework for IoT security.

We analyzed the characteristics of multiple knowledge bases of IoT security, and we proposed an IoT Security Threat Ontology (IoTSTO), which was inspired by UCO [12], IoTSec [19], and VulKG [43]. Furthermore, some concepts were extracted from these works, but with many details adjusted to make the ontology more suitable for knowledge bases. For example, we designed classes of tactics and techniques based on the ATT&CK matrix. The knowledge bases associated with these classes were not involved in the referenced ontology.

4.1. Classes and Attributes Analysis of IoTSTO

An ontology is a major component of semantic technology used in the modeling of data. In this paper, the ontology model is used as a bridge that generates services by conducting the knowledge reasoning for multi-source heterogeneous IoT security data. We apply an ontology-based reasoning, which is required for clear decision-making and a quick response to threats occurring continuously in an IoT environment. As shown in Figure 3, IoTSTO includes five top-level classes: *Platform*, *Vulnerability*, *Weakness*, *Attack Pattern*, and *Campaign*. The classification of classes is based on the previously mentioned knowledge base hierarchy, but it is different. *Platform* describes the scene information of the IoT security event, including the software, hardware, and operating system that may be affected by the vulnerability. Meanwhile, *Platform* also contains the product vendor and version information. *Status* is a subclass of the *Platform*, which describes the vulnerability of the affected platform. *Campaign* describes a set of malicious activities or attacks that occur against a set of specific targets over a period of time. *Campaign* can be expressed by the tactics, techniques, resources (tools, malware), groups that issued the malicious activities, and the mitigations that defend the system against this campaign.

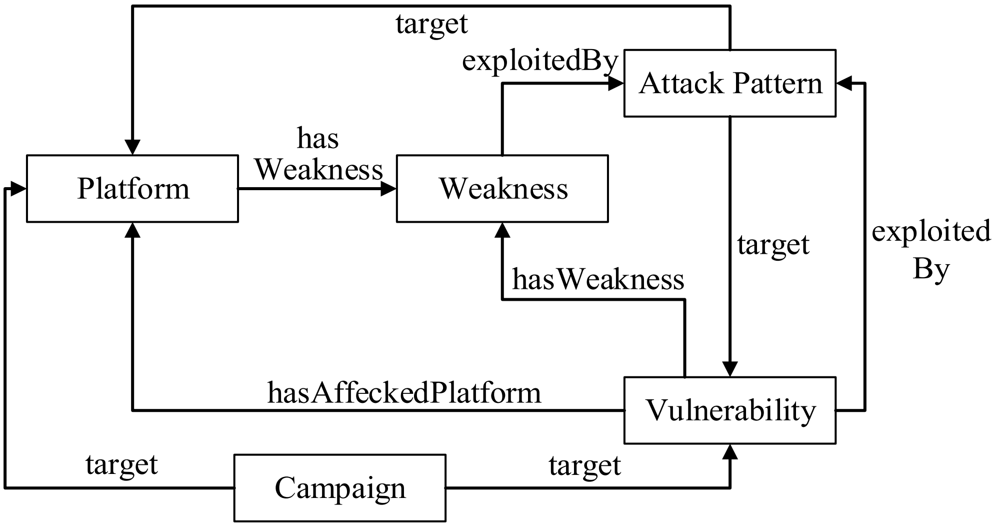


Figure 3. Top classes of IoTSTO.

Table 1 shows the level 2 and level 3 class definitions in the IoTSTO. The following paragraphs are based on the top-level classes of the IoTSTO and introduce various ontology descriptions of the sub-domains.

Table 1. Level 2 and level 3 ontology class definitions.

Level	Class	Level	Class
2	Platform	3	Status, Platform_Type, Product, Supply_Chain, Vendor, Version
2	Vulnerability	3	CVSS, Impact, Severity
2	Weakness	3	CWE_ID, Modes_Of_Introduction, Weakness_Type, Applicable_Platform, Weakness_Mitigation, Likelihood_Of_Exploit
2	Attack Pattern	3	CAPEC_ID, Attack_Likelihood, Attack_Mechanism, Attack_Pattern_Mitigation, Consequence, Prerequisite, Resources_Required
2	Campaign	3	Malware, Mitigation, Tactic, Technique, Threat_Group, Tool, Sub-Technique

4.1.1. Ontology Description of the Platform

The *Platform* class includes the software, hardware, and operating systems affected by threats in the IoT environment, and there are corresponding sub-classes and attributes to describe them. In this paper, we use the ontology language based on description logic (OWL DL) to represent the classes in the model. For example, using the description logic to describe the affected *Product*:

$$
\begin{aligned}
& Product \subseteq Platform \cap \\
& \quad \forall hasPlatformType.PT\ (Application \cup Hardware \cup OperatingSystem) \\
& \quad \exists hasSupplyChain.\ String \cap \\
& \quad \forall hasVendor.\ String \cap \\
& \quad \forall hasVersion.\ Version \cap \\
& \quad \forall hasStatus.\ Status\ (Normal\ Vulnerable \cup Serious\ Vulnerable \cup Critical\ Vulnerable)
\end{aligned}
$$

4.1.2. Ontology Description of the Vulnerability

In IoTSTO, the CVSS score describes the possible impact of the *Vulnerability* instance and also evaluates the severity of the *Vulnerability*. Taking vulnerability CVE-2017-7921 as an example—this vulnerability occurs when an application does not adequately or correctly authenticate users. Attackers can use token impersonation or session hijacking to escalate his or her privileges and gain access to sensitive information. CVE-2017-7921 is an instance of the *Vulnerability*, which is described as follows:

$Vulnerability$ (CVE-2017-7921)$\cap$
$\quad\forall hasSeverity.\ Severity(Critical) \cap$
$\quad\forall hasAttackVector.\ AV\ (Network) \cap$
$\quad\forall hasAttackComplexity.\ AC\ (Low) \cap$
$\quad\forall hasPrivilegesRequired.\ PR\ (None) \cap$
$\quad\forall hasUserInteraction.\ UI\ (None) \cap$
$\quad\forall hasScope.\ S\ (Changed) \cap$
$\quad\forall hasConfidentiality.\ C\ (High) \cap$
$\quad\forall hasIntegrity.\ I\ (High) \cap$
$\quad\forall hasAvailability.\ A\ (High) \cap$
$\quad\exists exploitedBy.\ AttackPattern\ (Token\ Impersonation \cup Session\ Hijacking)$

4.1.3. Ontology Description of the Weakness

In this paper, we introduce the *Modes_Of_Introduction* to describe how and when to import this weakness into the IoT environment. The subclass *Phase* identifies points that may be imported into the product life cycle. The subclass *Note* provides typical scenarios related to a specific import phase. CWE abstracts the weaknesses into 10 categories according to the relationship between them. *Weakness_Type* in this paper adopts this kind of classification method. *Improper Authentication* is an instance of the *Weakness*. When an attacker claims to have a given identity, the software does not prove or insufficiently proves that the claim is correct or authentic, which is described as follows:

$Improper\ Authentication \subseteq WeaknessType\ (Improper\ Access\ Control) \subseteq Weakness \cap$
$\quad\forall hasCWE\_ID.\ CWE\_ID$ (CWE-287) $\cap$
$\quad\forall hasApplicablePlatform.\ String \cap$
$\quad\forall hasWeaknessMitigation.\ String \cap$
$\quad\forall hasModesOfIntroduction.\ MOI\ (Phase \cap Note) \cap$
$\quad\forall hasLikelihoodOfExploit.\ LOE\ (High)$

4.1.4. Ontology Description of the Attack Pattern

CAPEC divides an attack pattern into nine categories according to the mechanism used by the attacker when exploiting the vulnerability. In this paper, the *Attack Mechanism* adopts this classification method, which includes nine subclasses that correspond to the classification given by CAPEC. *Session Hijacking* is an instance of the *Attack Pattern*; this type of attack involves an attacker that exploits the weaknesses in an application's use of sessions when performing authentication. The attacker is able to steal or manipulate an active session and use it to gain unauthorized access to the application, which is described as follows:

$AttackMechanism\ (Session\ Hijacking) \subseteq AttackPattern \cap$
$\quad\forall hasCAPEC\_ID.\ CAPEC\_ID$ (CAPEC-593) $\cap$
$\quad\forall hasAttackLikelihood.\ AL\ (High) \cap$
$\quad\forall hasAttackPatternMitigation.\ String \cap$
$\quad\forall hasConsequence.\ C\ (\exists Scope \cap \exists Impact) \cap$
$\quad\forall hasPrerequisite.\ String \cap$
$\quad\forall hasResourcesRequired.\ String$

4.1.5. Ontology Description of the Campaign

Compared with an attack, a campaign is a set of malicious activities or attacks that occur over a period of time against a specific set of targets. A *Campaign* class is a formal description of the tactics and techniques used by the attack group. In the ATT&CK

framework, each tactic contains multiple techniques, and each technique is composed of multiple sub-techniques. The ontology modeling of this paper refers to its structure. *Man-In-The-Middle* technique refers to the attackers positioning themselves between two or more networked devices to support follow-on behaviors such as Network Sniffing or Transmitted Data Manipulation. *Man-In-The-Middle* is an instance of the *Technique*, which is described as follows:

Technique (*Man-in-the-Middle*) $\subseteq$ *Campaign* $\cap$
$\forall$*belongToTactic. TA* (*Credential Access* $\cap$ *Collection*) $\cap$
$\forall$*hasSubTechnique. SubT* (*LLMNR/NBT-NS Poisoning and SMB Relay* $\cap$ *ARP Cache Poisoning*) $\cap$
$\forall$*hasMitigation. String* $\cap$
$\exists$*hasSoftware.* (*Tool* $\cap$ *Malware*) $\cap$
$\forall$*hasThreatGroup. Group*

4.2. Rule of Inference Design

In this paper, we choose Semantic Web Rule Language (SWRL) [44] to deal with the direct and indirect relationship of the ontology model to enhance the description ability of OWL. SWRL and OWL are based on the same description logic. SWRL has stronger expressive ability in solving the multi-level and complex interrelational reasoning and data value reasoning in ontology [45]. SWRL can directly use the relationships and vocabulary described in the ontology when writing rules of inference. Each SWRL rule is an OWL axiom in the ontology, and these new rules can also interact with the existing axioms in the ontology. The form of the SWRL is given as follows:

$$A_1, \ldots, A_m \rightarrow B_1, \ldots, B_n$$

The commas on both sides of the arrows indicate conjunctions, which can be written as a conjunctive form and a disjunctive form when describing complex logical relationships. $A_1, \ldots, A_m \rightarrow B_1, \ldots, B_n$ can be expressed as $C(x)$, $P(x, y)$, or (x, y). Here, C is an OWL description, P is an OWL property, and x and y can be datalog variables, OWL instances, or OWL data values. The rule of inference can discover new implicit knowledge from explicit knowledge. The following example shows the usage of the SWRL rule. Figure 4 reveals the new relationship according to the inference rule.

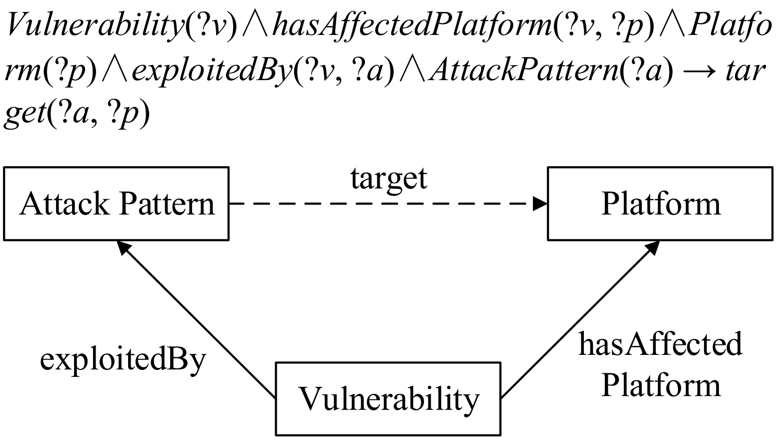

Vulnerability(?v)⋀hasAffectedPlatform(?v, ?p)⋀Platfo
rm(?p)⋀exploitedBy(?v, ?a)⋀AttackPattern(?a) → tar
get(?a, ?p)
Attack Pattern
target
Platform
exploitedBy
Vulnerability
hasAffected
Platform

Figure 4. Examples of the SWRL inference rule and representation.

In the inference rule above, the object properties *hasAffectedPlatform*(?*v*, ?*p*) provides the relationship linkage between the class *Vulnerability*(?*v*) and the class *Platform*(?*p*). The object properties *exploitedBy*(?*v*, ?*a*) provides the relationship linkage between the class

Vulnerability(?*v*) and the class *AttackPattern*(?*a*). This kind of relational linkage can infer implicit facts from the object properties *target*(?*a*, ?*p*) and existing knowledge.

The SWRL cannot make the OWL query. Therefore, we use Semantic Query-Enhanced Web Rule Language (SQWRL) to perform knowledge retrieval on the integrated ontology model. SQWRL is an extension of SWRL, and it can be used in conjunction with SWRL to transform existing rules into a pattern-matching mechanism. It allows for the query and retrieval of implicit knowledge inferred from OWL classes, OWL object properties, OWL data properties, and OWL individuals. The example in Section 4 shows the specific usage of inference rule in the multi-source knowledge inference model of IoT security.

5. Examples and Evaluation

In this section, we demonstrate several scenarios to further illustrate the feasibility and effectiveness of the proposed model. As for the coalescence and modeling of the multi-source information security knowledge base, the first subsection provides a linkage example to demonstrate the ability of the integrated graph data in order to provide context semantic information. The second subsection focuses on the design of inference rules based on the multi-source knowledge of IoT security. Various hardware and software in the IoT correspond to instances in the ontology. The security status of the IoT environment is reflected in the ontology. In this paper, instances in the IoT environment are mapped in the ontology, and the ability of the model to construct threat assessments of the IoT environment is demonstrated by its design of inference rules.

5.1. Linkage Example and Feasibility Analysis

This paper integrates multiple cyber security knowledge bases to provide a seamless set of paths that connect them. In order to demonstrate the feasibility and advantages of this method, this section uses a linkage query as an example, such as that of “video surveillance devices” to “Privilege Escalation” and “Defense Evasion” tactics. The entries in the linkage are as follows:

- *Tactic* (TA0004) *Privilege Escalation*: This adversary is trying to gain higher-level permissions. Privilege Escalation consists of techniques that adversaries use to gain higher-level permissions in a system or network. Adversaries can often enter and explore a network with unprivileged access but require elevated permissions to follow through on their objectives. Obtaining an account that is necessary for attackers to achieve their goals of gaining access to a specific system or performing a specific authorized operation can also be considered a privilege escalation. Common approaches are taking advantage of system weaknesses, misconfigurations, and vulnerabilities.
- *Tactic* (TA0005) *Defense Evasion*: This adversary is trying to avoid being detected. Defense Evasion consists of techniques that adversaries use to avoid detection throughout their compromise. Adversaries also leverage and abuse trusted processes to hide and masquerade their malware.
- *Technique* (T1134) *Access Token Manipulation*: Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. The operation system, such as Windows, uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it is the child of a different process or belongs to someone other than the user that started the process.
- *Attack Pattern* (CAPEC-633) *Token Impersonation*: An adversary exploits a weakness in authentication to create an access token that impersonates a different entity, and then associates a process to that that impersonated token. Attackers can use this operation to use tokens to verify identity and take actions based on that identity.
- *Weakness* (CWE-287) *Improper Authentication*: When an actor claims to have a given identity, the platform does not prove or insufficiently proves that the claim is correct.
- *Vulnerability*: CVE-2017-7921. The improper authentication vulnerability occurs when an application does not adequately or correctly authenticate users. This may allow

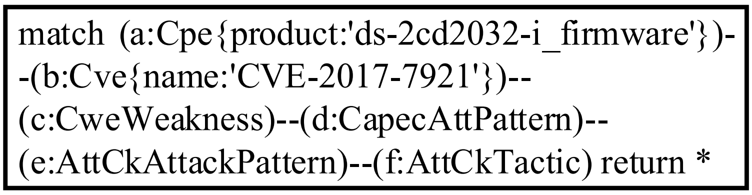
a malicious user to escalate his or her privileges on the system and gain access to sensitive information.

- *Affected platform* and CPE: “cpe:2.3:o: hikvision:ds-2cd2032-i_firmware:-:*:*:*:*:*:*:*′,” According to the CPE entry, the affected platforms are Hikvision video surveillance devices with firmware version DS-2CD2032-I.

The description of the linkage above is based on tactic from the perspective of attackers, assuming that the attacker’s goal is to obtain higher-level permissions without being detected. That is, by manipulating the access token to run as different users or in different systems in order to perform operations and bypass access control. Attackers can exploit the vulnerability of the Hikvision video surveillance devices, whose firmware version is DS-2CD2032-I, to simulate the access tokens of different entities through the weaknesses in authentication, and then escalate the privilege to obtain sensitive information and control this video surveillance device.

On the other hand, the description of the linkage is based on the affected platform from the perspective of defenders. If there are Hikvision video surveillance devices with the firmware version DS-2CD2032-I in cyberspace, network administrators need to be vigilant of attackers simulating access tokens through weaknesses in authentication, bypassing access control, and achieving privilege escalation. Network administrators can restrict permissions of users and user groups who cannot create tokens, or define token permissions for specific users in order to manage and restrict token creation. At the same time, network administrators can restrict users and accounts to the minimum privileges they need. They can reduce the path where attackers can bypass access control and narrow the possible attack surface to mitigate threats.

As shown in the example above, researchers who use this method to integrate multi-source knowledge bases can query context information from different knowledge bases with any entry in the link set given. In fact, this method can traverse the knowledge graph to achieve more powerful query functions according to specific query requirements. The preprocessed graph data is stored in Neo4j. We used the query language Cypher and queried the Hikvision video surveillance device with the firmware version DS-2CD2032-I and the vulnerability entry CVE-2017-7921. The Cypher query statement is shown in Figure 5.



```
match (a:Cpe{product:'ds-2cd2032-i_firmware'})-
-(b:Cve{name:'CVE-2017-7921'})--
(c:CweWeakness)--(d:CapecAttPattern)--
(e:AttCkAttackPattern)--(f:AttCkTactic) return *
```

Figure 5. Cypher query for CVE-2017-7921.

Figure 6 shows the visualization of the query results. After the vulnerability CVE-2017-7921 is associated with the weakness CWE-287 Improper Authentication, four linkages associated with CWE-287 are traversed in the graph database, with the sample linkage query above also being among them. The context information contains several public security information resources that have been published and associated. The integration of multi-source knowledge existing in a specific IoT environment can improve the analysis ability and comprehensibility of CTI.

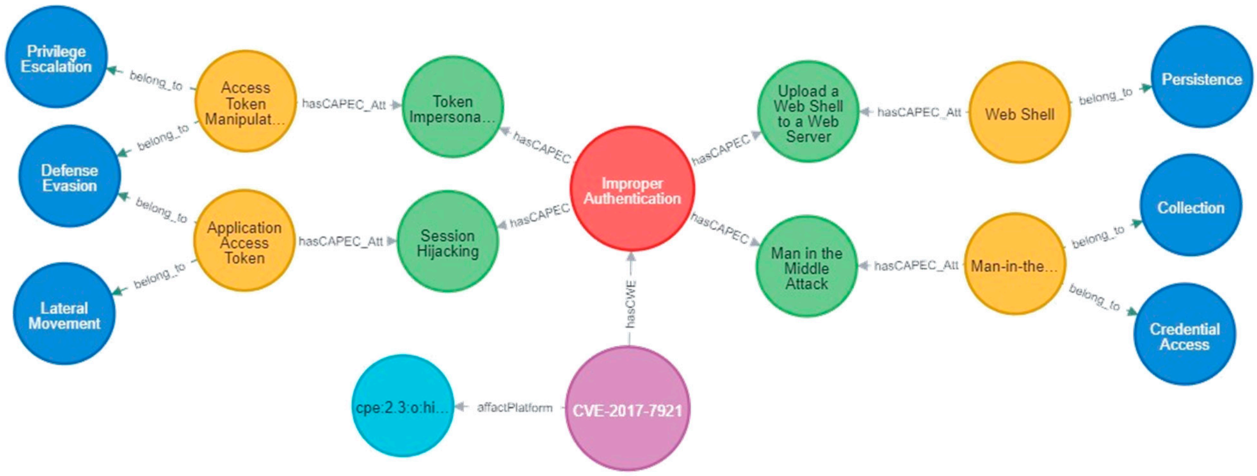


Figure 6. Cypher query result visualization for CVE-2017-7921.

5.2. Inference Rules Based on Multi-Source Knowledge of IoT Security

This paper constructs a sample scenario for IoT testing, as shown in Figure 7. All components are instantiated in the IoTSTO. There is a video surveillance device collaboration group in the IoT, and devices are all connected to the Internet. There are multiple Hikvision video surveillance devices. The firmware versions are DS-2CD2032-I, DS-2CD2432-IW, and DS-7204HGHI-F1. The manager of the video surveillance device collaboration group uses the Ivms-4200 network video surveillance software. The existing vulnerabilities in the environment and their CVSS scores and severity are shown in Table 2.

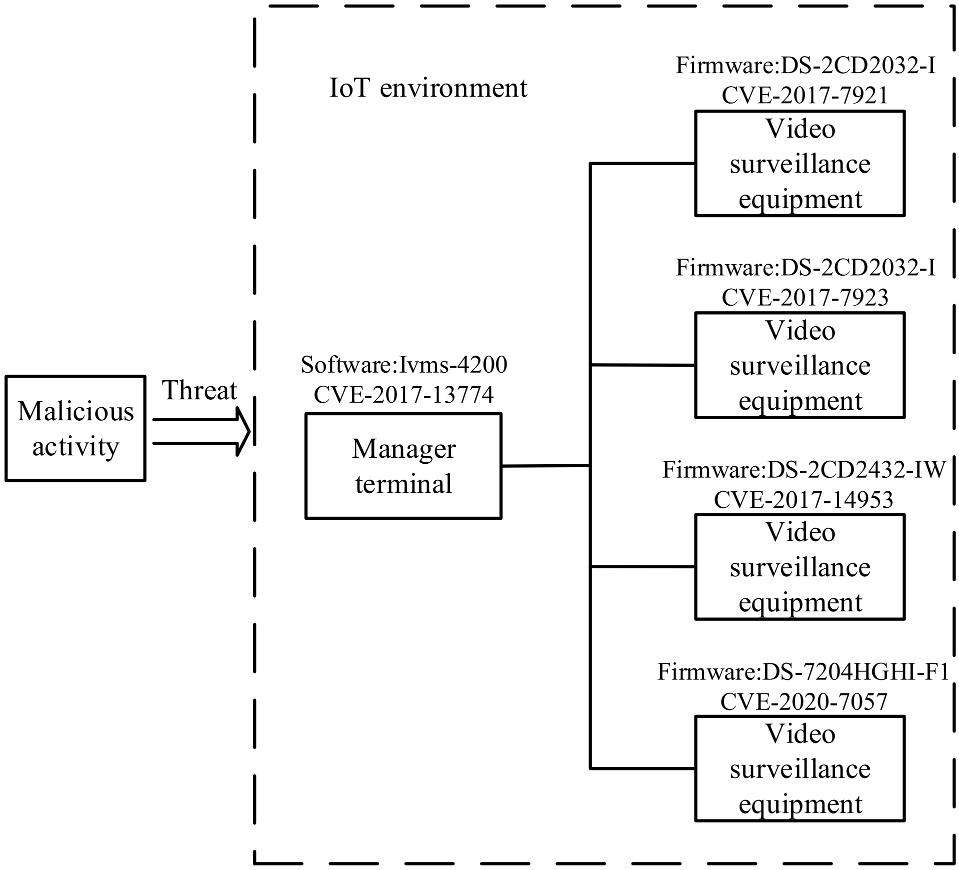


Figure 7. IoT context example.

Table 2. Vulnerability, CVSS score and severity.

IoT Platform	Vulnerability	CVSS V3	Severity
DS-2CD2032-I	CVE-2017-7921	10.0	Critical
DS-2CD2032-I	CVE-2017-7923	8.8	High
Ivms-4200	CVE-2017-13774	7.8	High
DS-2CD2432-IW	CVE-2017-14953	6.5	Medium
DS-7204HGHI-F1	CVE-2020-7057	5.3	Medium

These components are instantiated in the IoTSTO. For this paper, we designed the five inference rules to infer the security status in the IoT environment, as shown in Figure 8.

Rule-1:
$Vulnerability(?v) \wedge hasSeverity(?v,\ ?se) \wedge Severity(?se,\ CriticalSeverity) \wedge exploitedBy(?v,\ ?a) \wedge AttackPattern(?a) \rightarrow CriticalVulnerability(?v)$
Rule-2:
$Vulnerability(?v) \wedge hasSeverity(?v,\ ?se) \wedge Severity(?se,\ HighSeverity) \wedge exploitedBy(?v,\ ?a) \wedge AttackPattern(?a) \rightarrow SeriousVulnerability(?v)$
Rule-3:
$Vulnerability(?v) \wedge hasSeverity(?v,\ ?se) \wedge Severity(?se,\ MediumSeverity) \wedge exploitedBy(?v,\ ?a) \wedge AttackPattern(?a) \rightarrow NormalVulnerability(?v)$
Rule-4:
$Vulnerability(?v) \wedge hasSeverity(?v,\ ?se) \wedge Severity(?se,\ LowSeverity) \wedge exploitedBy(?v,\ ?a) \wedge AttackPattern(?a) \rightarrow NormalVulnerability(?v)$
Rule-5:
$Platform(?p) \wedge hasVulnerability(?p,\ ?v) \wedge Vulnerability(?v) \wedge hasSeverity(?v,\ ?se) \wedge Severity(?se) \wedge Status(?st) \wedge affect(?se,\ ?st) \rightarrow hasStatus(?p,\ ?st)$

Figure 8. Reasoning the security status of the IoT environment.

Rule-1 to Rule-4 identify the *Severity* based on the known CVSS score of the *Vulnerability*. The semantics of Rule-5 is that there are *Platforms* affected by the *Vulnerability* in the IoT environment. When the *Severity* of the *Vulnerability* is "High" or "Critical", the *Platform* affected by the *Vulnerability* in the system is *Critical Vulnerable* to malicious activities. The system infers the *Critical Vulnerable* areas of the *Platform* and provides support for security analysts to assess the *Severity* of threats to the IoT environment. Based on the inference rules above, we designed Rule-6 to realize automatically inferred *Mitigations* for *Critical Vulnerable Platforms*, as shown in Figure 9.

Rule-6:
$Platform(?p) \wedge hasStatus(?p,\ ?st) \wedge Status(?st) \wedge hasVulnerability(?p,\ ?v) \wedge Vulnerability(?v) \wedge AttackPattern(?a\ ) \wedge target(?a,\ ?v) \wedge mapping(?a,\ ?t) \wedge Technique(?t) \wedge Mitigation(?m) \rightarrow mitigated(?p,\ ?m)$

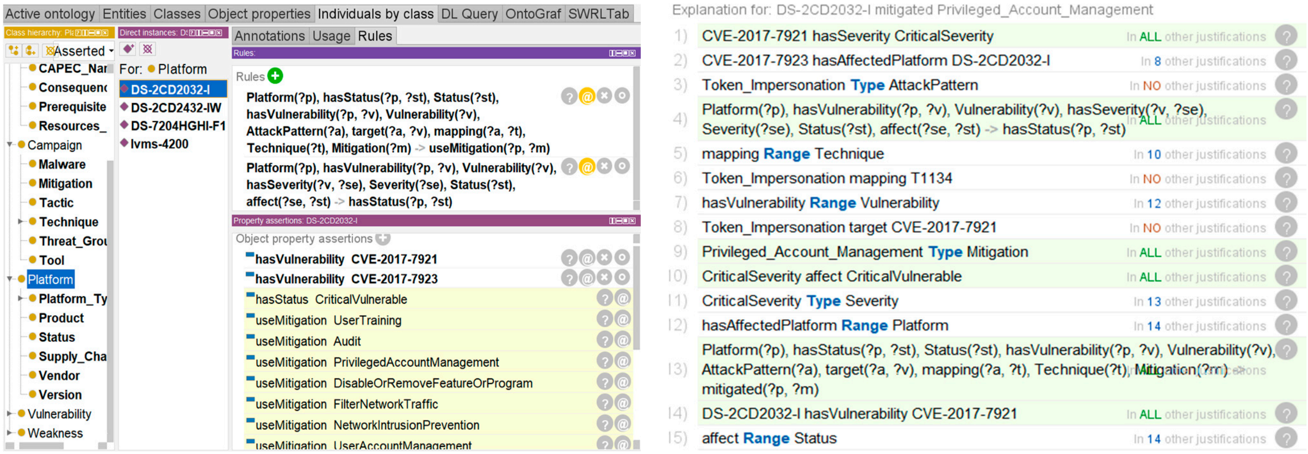
Figure 9. Reasoning mitigation.

The semantics of this rule is that when the location of the *Critical Vulnerable* in the IoT environment is known, the *Attack Pattern* that may be subject to the *Vulnerability* in the *Platform* is mapped to the *Technique*. Then, network managers use the known threat events in the knowledge base to analyze the malicious activities and tactics associated with the specific *Technique*. Finally, the inference engine automatically infers the *Mitigations* that can be adopted in the system.

After constructing the rules above, the Pellet reasoner is initiated for reasoning. According to the DS-2CD2032-I entity, new implicit facts are inferred. The *Status* of DS-

2CD2032-I is classified as *Critical Vulnerable* and is associated with *Mitigation* through the object property *useMitigation*. The reasoning result is shown in Figure 10a.

Through the interpretation function of Protégé, we showed the inference process of the security status of the IoT environment, and demonstrated the process of Rule-6 of automating inference mitigations inferred by the implicit facts. The details are shown in Figures 10b and 11.



Platform(?p), hasStatus(?p, ?st), Status(?st), hasVulnerability(?p, ?v), Vulnerability(?v), AttackPattern(?a), target(?a, ?v), mapping(?a, ?t), Technique(?t), Mitigation(?m) -> useMitigation(?p, ?m)

Platform(?p), hasVulnerability(?p, ?v), Vulnerability(?v), hasSeverity(?v, ?se), Severity(?se), Status(?st), affect(?se, ?st) -> hasStatus(?p, ?st)

hasVulnerability CVE-2017-7921
hasVulnerability CVE-2017-7923
hasStatus CriticalVulnerable
useMitigation UserTraining
useMitigation Audit
useMitigation PrivilegedAccountManagement
useMitigation DisableOrRemoveFeatureOrProgram
useMitigation FilterNetworkTraffic
useMitigation NetworkIntrusionPrevention
useMitigation UserAccountManagement

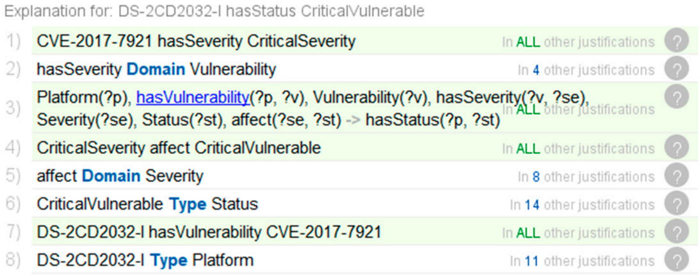
(**a**)

Explanation for: DS-2CD2032-I mitigated Privileged_Account_Management

1) CVE-2017-7921 hasSeverity CriticalSeverity — In ALL other justifications
2) CVE-2017-7923 hasAffectedPlatform DS-2CD2032-I — In 8 other justifications
3) Token_Impersonation Type AttackPattern — In NO other justifications
4) Platform(?p), hasVulnerability(?p, ?v), Vulnerability(?v), hasSeverity(?v, ?se), Severity(?se), Status(?st), affect(?se, ?st) -> hasStatus(?p, ?st) — In ALL other justifications
5) mapping Range Technique — In 10 other justifications
6) Token_Impersonation mapping T1134 — In NO other justifications
7) hasVulnerability Range Vulnerability — In 12 other justifications
8) Token_Impersonation target CVE-2017-7921 — In NO other justifications
9) Privileged_Account_Management Type Mitigation — In ALL other justifications
10) CriticalSeverity affect CriticalVulnerable — In ALL other justifications
11) CriticalSeverity Type Severity — In 13 other justifications
12) hasAffectedPlatform Range Platform — In 14 other justifications
13) Platform(?p), hasStatus(?p, ?st), Status(?st), hasVulnerability(?p, ?v), Vulnerability(?v), AttackPattern(?a), target(?a, ?v), mapping(?a, ?t), Technique(?t), Mitigation(?m) -> mitigated(?p, ?m)
14) DS-2CD2032-I hasVulnerability CVE-2017-7921 — In ALL other justifications
15) affect Range Status — In 14 other justifications

(**b**)

Figure 10. (**a**) Results from the application of inference rules; (**b**) The joint inference process.



Explanation for: DS-2CD2032-I hasStatus CriticalVulnerable

1) CVE-2017-7921 hasSeverity CriticalSeverity — In ALL other justifications
2) hasSeverity Domain Vulnerability — In 4 other justifications
3) Platform(?p), hasVulnerability(?p, ?v), Vulnerability(?v), hasSeverity(?v, ?se), Severity(?se), Status(?st), affect(?se, ?st) -> hasStatus(?p, ?st) — In ALL other justifications
4) CriticalSeverity affect CriticalVulnerable — In ALL other justifications
5) affect Domain Severity — In 8 other justifications
6) CriticalVulnerable Type Status — In 14 other justifications
7) DS-2CD2032-I hasVulnerability CVE-2017-7921 — In ALL other justifications
8) DS-2CD2032-I Type Platform — In 11 other justifications

Figure 11. Inference process to infer the security status.

The detailed inference processes are described as follows:

1. A video surveillance device with firmware DS-2CD2032-I is deployed in the IoT. According to the explicit knowledge in the knowledge bases CVE and NVD, this video surveillance device has a CVE-2017-7921 vulnerability, and the *Severity* is *CriticalSeverity*;
2. System classifies the *Status* describing the vulnerability of the devices as *CriticalVulnerable*, which is based on the *Severity* of the *Vulnerability* associated with the video surveillance device
3. According to the explicit knowledge in the knowledge base CAPEC and the ATT&CK matrix, the *Attack Pattern* Token Impersonation is mapped to *Technique* T1134. System analyzes related threat events, which can infer appropriate *Mitigations* to mitigate threat activities that may be generated by adversaries.

The inference engine can separate the IoT nodes that are subjected to specific threats. The weaknesses and threats in the IoT environment are random and large in number. However, the targets of some threats are specific, and the characteristics of threat activities are obvious [46]. After perceiving the distribution of vulnerability in the IoT environment, the system combines the context information of threat intelligence to separate high-risk nodes and prioritize the processing of specific threats, which will greatly reduce spatio-temporal consumption and increase the speed of the response to threats [47,48]. For example, as shown in Figure 12, Rule-7 checks whether there are weaknesses of *Improper Authentication* in the IoT environment. The inference engine separates high-risk nodes that may be used by attackers in order to use the technique *Access Token Manipulation* to complete *Token Impersonation* of the IoT environment. In addition, network managers only need to add corresponding instances to the data layer in order to update the *Platforms* in the IoT environment, while IoTSTO and rule of inference can easily infer the security *status* in the IoT.

Rule-7:
Platform(*?p*)∧*hasVulnerability*(*?p*, *?v*)∧*Vulnerability*(*?v*)∧*hasStatus*(*?p*, *CriticalVulnerab*
le)∧*hasWeakness*(*?p*, *Improper Authentication*)∧*target*(*Token Impersona tion*, *?p*)∧*Attack*
Pattern(*?a*)∧*exploitedBy*(*?v*, *?a*)∧*ma pping*(*?a*, *Access Token Manipulation*) → *sqwl*:*select*
(*?p*, *?v*)

Figure 12. Separate critical vulnerable platforms with specific patterns.

The example verifies the ability of our method to discover and automatically defend against threats in a heterogeneous IoT environment. If the system has vulnerabilities that are exploited by certain attack patterns, tactics, or techniques, the IoTSTO will perceive the critical-risk locations of system security through preset inference rules. Security managers can automatically infer the available mitigations and separate IoT nodes with specific threats from the constructed multi-source knowledge base. This method provides support for security managers to perceive the overall security situation and deploy the appropriate security solutions. The most important thing is to design the inference rules. Inference rules should be designed according to the actual IoT environment, and the characteristics of the complex IoT environment need to be accumulated and analyzed.

The current cyber security ontology models focus on different scopes of information. Some models focus on the integration of IoT assets, vulnerabilities, and weaknesses [19,43,49], and some models focus on modeling attack in the IoT environment [12,18,20,39,50,51]. Our proposed ontology model focuses on the information on IoT assets, vulnerabilities, weaknesses, attack patterns, techniques, and tactics, which gives a holistic view of the cyber security situation and is more comprehensive than the other models. Table 3 compares IoTSTO and other ontology models to show the scope of the knowledge domains involved in each cyber security ontology.

Table 3. Knowledge domain scope of several related cybersecurity ontology models.

	Platform	Vulnerability	Weakness	Attack Pattern	Technique	Tactic
UCO [12]		√		√	√	
MALOnt [18]		√		√	√	
IoTSec [19]	√	√	√			
PIoTCO [20]	√			√		
NSSA [39]	√	√		√		
VulKG [43]	√	√	√			
VCO [49]	√	√	√	√		
SVO [50]		√	√	√		
SKO [51]		√		√		
IoTSTO	√	√	√	√	√	√

In this paper, IoTSTO focused on the affected platform, vulnerability, weakness, attack pattern, tactic, and technique used by attackers in the IoT, and provided a broader perception of the security status of the IoT. The scalability of IoTSTO is sufficient to accommodate the rapid transformation of the IoT architecture. Managers can define relevant inference rules based on the characteristics of the observed IoT environment in order to meet the adaptability of the model to the actual IoT environment and enhance the inference ability of the ontology.

6. Conclusions

In this paper, we integrated vulnerabilities, weaknesses, affected platforms and tactics, attack techniques, and attack patterns into a coherent set of links. This method enriched the context information of the network security knowledge base, which improve the analytical ability and comprehensibility of the CTI. We resolved the semantic heterogeneity problem by facilitating the formalization of knowledge in the IoT domain. In addition, we proposed an IoT security threat ontology model to describe the elements of IoT security threats, and we used a Pellet reasoner and an inference rule to perceive threats in a complex heterogeneous environment.

However, our work is not enough to monitor the overall security of the IoT. Subsequently, we will add cyber supply chain information to the ontology model. For example, we can add the supply chain information of intelligent manufacturing devices, such as third party vendors, suppliers, inbound supply chain, outbound supply chain, and other status information to the ontology model. In this way, it will not only detect the threat, but will also be able to detect the novel supply chain attacks by fusing the supply chain information of the intelligent manufacturing physical system with the cyber security information.

Author Contributions: Conceptualization, S.Z., G.B., H.L., P.L., M.Z. and S.L.; methodology, S.Z. and G.B.; software, G.B.; validation, S.Z., G.B., H.L. and P.L.; formal analysis, G.B.; investigation, S.Z. and G.B.; resources, G.B. and M.Z.; data curation, G.B. and M.Z.; writing—original draft preparation, G.B.; writing—review and editing, S.Z., G.B., H.L., P.L. and S.L.; visualization, G.B.; supervision, S.Z. and H.L.; project administration, S.Z. and G.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Program Research Fund of Higher Education of Henan, China, grant number No. 18B520044, No. 19A520048, and the Science and Technique Foundation of Henan, China, grant number No. 182102210526.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Borgia, E. The internet of things vision: Key features, applications and open issues. *Comput. Commun.* **2014**, *1*, 1–31. [CrossRef]
2. Mohammad, S.J.; Jessica, P.K.; Michael, S.; Stuart, M. The internet of things promises new benefits and risks: A systematic analysis of adoption dynamics of IoT products. *IEEE Secur. Priv.* **2019**, *17*, 39–48.
3. CISA: Advanced Persistent Threat Compromise of Government Agencies, Critical Infrastructure, and Private Sector Organizations. Available online: https://us-cert.cisa.gov/ncas/alerts/aa20-352a (accessed on 27 October 2021).
4. SECURELIST by Kaspersky: Popular Server Management Software Hit in Supply Chain Attack. Available online: https://securelist.com/shadowpad-in-corporate-networks/81432/ (accessed on 27 October 2021).
5. Song, F.; Ai, Z.; Zhou, Y.; You, I.; Zhang, H. Smart collaborative automation for receive buffer control in multipath industrial networks. *IEEE Trans. Ind. Inform.* **2020**, *16*, 1385–1394. [CrossRef]
6. Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet Things J.* **2017**, *4*, 1125–1142. [CrossRef]
7. Schinagl, S.; Schoon, K.; Paans, R. A framework for designing a security operations centre (SOC). In Proceedings of the 2015 48th Hawaii International Conference on System Sciences (HICSS), Washington, DC, USA, 5–8 January 2015; pp. 2253–2262.
8. Antonio, L.M.; Fabio, M.; Francesco, M.; Andrea, S.; Mina, S. D-BRIDEMAID: A distributed framework for collaborative and dynamic analysis of android malware. *JoWUA* **2020**, *11*, 1–28.
9. Zeshan, F.; Ahmad, A.; Abdel-Aty, A.-H.; Algarni, F.; Mahmoud, E.E.; Ahmad, A. A hybrid semantic knowledge integration and sharing approach for distributed smart environments. *Sensors* **2020**, *20*, 5918. [CrossRef]
10. Clayton, J.; Bishal, K.; Ram, B.B.; Tenzin, D. Towards detecting and classifying malicious URLs using deep learning. *JoWUA* **2020**, *11*, 31–48.

11. Davide, C.; Luca, V.; Andrea, R.; Alessio, M.; Luca, C. Fine-hearing Google Home: Why silence will not protect your privacy. *JoWUA* **2020**, *11*, 35–53.
12. Syed, Z.; Padia, A.; Finin, T.; Mathews, L.; Joshi, A. UCO: A unified cybersecurity ontology. In Proceedings of the 2016 AAAI Workshop on Artificial Intelligence for Cyber Security, Menlo Park, CA, USA, 12 February 2016.
13. Abbas, S.G.; Vaccari, I.; Hussain, F.; Zahid, S.; Fayyaz, U.U.; Shah, G.A.; Bakhshi, T.; Cambiaso, E. Identifying and mitigating phishing attack threats in IoT use cases using a threat modelling approach. *Sensors* **2021**, *21*, 4816. [CrossRef]
14. Huang, K.; Lian, Y.; Feng, D.; Zhang, H.; Liu, Y.; Ma, X. Cyber security threat intelligence sharing model based on blockchain. *J. Comput. Res. Dev.* **2020**, *57*, 836–846.
15. Kiesling, E.; Ekelhart, A.; Kurniawan, K.; Ekaputra, F. The SEPSES knowledge graph: An integrated resource for cybersecurity. In Proceedings of the Semantic Web—ISWC 2019, Auckland, New Zealand, 26–30 October 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 198–214.
16. Tian, J.; Tian, Z.; Qi, W.; Hao, H.; Li, R.; Li, X.; Qiao, H.; Xue, H. Threat propagation based security situation quantitative assessment in multi-node network. *J. Comput. Res. Dev.* **2017**, *54*, 731–741.
17. Jia, Y.; Qi, Y.L.; Shang, H.J.; Li, A. A practical approach to constructing a knowledge graph for cybersecurity. *Engineering* **2018**, *4*, 53–60. [CrossRef]
18. Rastogi, N.; Dutta, S.; Zaki, M.J.; Gittens, A.; Aggarwal, C. MALOnt: An ontology for malware threat intelligence. *arXiv* **2020**, arXiv:2006.11446.
19. Mozzaquatro, B.; Goncalves, R.J.; Agostinho, C. Towards a reference ontology for security in the internet of things. In Proceedings of the 2015 IEEE International Workshop on Measurements & Networking, Coimbra, Portugal, 12–13 October 2015; pp. 1–6.
20. Choi, C.; Choi, J. Ontology-based security context reasoning for power IoT-cloud security service. *IEEE Access* **2019**, *7*, 110510–110517. [CrossRef]
21. Si, C.; Zhang, H.Q.; Wang, Y.W.; Yang, Y. Research on network security situational elements knowledge base model based on ontology. *Comput. Sci.* **2015**, *42*, 173–177.
22. Li, T.; Lin, J.C.; Hu, A.Q. Ontology model based on security parameters capturing process for network systems. *Chin. J. Netw. Inf. Secur.* **2017**, *3*, 39–45.
23. Mozzaquatro, B.; Agostinho, C.; Goncalves, D.; Martins, J.; Goncalves, R.J. An ontology-based cybersecurity framework for the internet of things. *Sensors* **2018**, *18*, 3053. [CrossRef] [PubMed]
24. Igor, T.; Petra, G. Towards the open ontology for IoT ecosystem's security. In Proceedings of the 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 28 September 2020.
25. MITRE. Common Vulnerabilities and Exposure. Available online: https://cve.mitre.org/ (accessed on 15 December 2020).
26. NIST. National Vulnerability Databased. Available online: https://nvd.nist.gov (accessed on 15 December 2020).
27. MITRE. Common Weakness Enumeration. Available online: https://cwe.mitre.org/ (accessed on 15 December 2020).
28. MITRE. Common Attack Pattern Enumeration and Classification. Available online: https://capec.mitre.org/ (accessed on 15 December 2020).
29. NIST. Common Platform Enumeration. Available online: https://nvd.nist.gov/Products/CPE (accessed on 15 December 2020).
30. MITRE. ATT&CK Matrix for Enterprise. Available online: https://attack.mitre.org/ (accessed on 15 December 2020).
31. FIRST. Common Vulnerability Scoring System. Available online: https://www.first.org/cvss/ (accessed on 15 December 2020).
32. MITRE. Common Weakness Enumeration. Available online: https://cwe.mitre.org/about/index.html (accessed on 15 December 2020).
33. Cheng, X.; Zhang, J.; Chen, B. Cyber situation comprehension for IoT systems based on APT alerts and logs correlation. *Sensors* **2019**, *19*, 4045. [CrossRef]
34. Dong, C.; Jiang, B.; Lu, Z.G.; Liu, B.; Li, N.; Ma, P. Knowledge graph for cyberspace security intelligence: A survey. *J. Cyber Secur.* **2020**, *5*, 56–76.
35. Abhishta, A.; Wouter, V.H.; Marianne, J.; Lambert, J.M.N.; Reinoud, J. Why would we get attacked? An analysis of attacker's aims behind DDoS attacks. *JoWUA* **2020**, *11*, 3–22.
36. Yang, Z.M.; Li, Q.; Liu, J.R. Research of threat intelligence sharing and using for cyber attack attribution. *J. Inf. Secur. Res.* **2015**, *1*, 37–42.
37. Song, F.; Zhu, M.; Zhou, Y.; You, I.; Zhang, H. Smart collaborative tracking for ubiquitous power IoT in edge-cloud interplay domain. *IEEE Internet Things J.* **2020**, *7*, 6046–6055. [CrossRef]
38. Kasturi, G.S.; Jain, A.; Singh, J. Detection and classification of radio frequency jamming attacks using machine learning. *JoWUA* **2020**, *11*, 49–62.
39. Xu, G.Q.; Cao, Y.; Ren, Y.Y.; Li, X.; Feng, Z. Network security situation awareness based on semantic ontology and user-defined rules for internet of things. *IEEE Access* **2017**, *5*, 21046–21056. [CrossRef]
40. Lee, O.J.; Nguyen, H.L.; Jung, J.E.; Um, T.; Lee, H. Towards ontological approach on trust-aware ambient services. *IEEE Access* **2017**, *5*, 1589–1599. [CrossRef]
41. Li, J.; Li, X.L.; Yang, B.; Sun, X. Segmentation-based image copy-move forgery detection scheme. *IEEE Trans. Inf. Forensics Secur.* **2017**, *10*, 507–518.
42. W3C Semantic Web. Available online: https://www.w3.org/OWL/ (accessed on 17 March 2021).

43. Qin, S.Z.; Chow, K.P. Automatic analysis and reasoning based on vulnerability knowledge graph. In *Cyberspace Data and Intelligence, and Cyber-Living, Syndrome, and Health*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 3–19.
44. Ian, H.; Peter, F.P.; Harold, B.; Said, T.; Benjamin, G.; Mike, D. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Available online: http://www.daml.org/rules/proposal/ (accessed on 15 April 2021).
45. Peng, C.; Goswami, P. Meaningful integration of data from heterogeneous health services and home environment based on ontology. *Sensors* **2019**, *19*, 1747. [CrossRef]
46. Song, F.; Li, L.; You, I.; Zhang, H. Enabling heterogeneous deterministic networks with smart collaborative theory. *IEEE Netw.* **2021**, *35*, 64–71. [CrossRef]
47. Song, F.; Ai, Z.; Zhang, H.; You, I.; Li, S. Smart collaborative balancing for dependable network components in cyber-physical systems. *IEEE Trans. Industr. Inform.* **2021**, *17*, 6916–6924. [CrossRef]
48. Song, F.; Zhou, Y.; Wang, Y.; Zhao, T.; You, I.; Zhang, H. Smart collaborative distribution for privacy enhancement in moving target defense. *Inform. Sci.* **2019**, *479*, 593–606. [CrossRef]
49. Shang, H.J.; Rong, J.; Li, A.P.; Wei, W. A framework to construct knowledge base for cyber security. In Proceedings of the 2017 IEEE Second International Conference on Data Science in Cyberspace, Shenzhen, China, 26–29 June 2017; pp. 242–248.
50. Zhu, L.N.; Zhang, Z.C.; Xia, G.E.; Jiang, C. Research on vulnerability ontology model. In Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 24–26 May 2019; pp. 657–661.
51. Qi, Y.L.; Jiang, R.; Jia, Y.; Li, A. Attack analysis framework for cyber-attack and defense test platform. *Electronics* **2020**, *9*, 1413. [CrossRef]