Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.040964

ARTICLE

Threat Modeling and Application Research Based on Multi-Source Attack and Defense Knowledge

Shuqin Zhang, Xinyu Su*, Peiyu Shi, Tianhui Du and Yunfei Han

School of Computer Science, Zhongyuan University of Technology, Zhengzhou, China

*Corresponding Author: Xinyu Su. Email: xinyu.su@zut.edu.cn

Received: 06 April 2023 Accepted: 25 July 2023 Published: 31 October 2023

ABSTRACT

Cyber Threat Intelligence (CTI) is a valuable resource for cybersecurity defense, but it also poses challenges due to its multi-source and heterogeneous nature. Security personnel may be unable to use CTI effectively to understand the condition and trend of a cyberattack and respond promptly. To address these challenges, we propose a novel approach that consists of three steps. First, we construct the attack and defense analysis of the cybersecurity ontology (ADACO) model by integrating multiple cybersecurity databases. Second, we develop the threat evolution prediction algorithm (TEPA), which can automatically detect threats at device nodes, correlate and map multi-source threat information, and dynamically infer the threat evolution process. TEPA leverages knowledge graphs to represent comprehensive threat scenarios and achieves better performance in simulated experiments by combining structural and textual features of entities. Third, we design the intelligent defense decision algorithm (IDDA), which can provide intelligent recommendations for security personnel regarding the most suitable defense techniques. IDDA outperforms the baseline methods in the comparative experiment.

KEYWORDS

Multi-source data fusion; threat modeling; threat propagation path; knowledge graph; intelligent defense decision-making

1 Introduction

The progressive expansion of the Internet into various areas, including e-commerce, education, and online media, has resulted in a sharp rise in threat events. A critical resource for comprehending threats is Cyber Threat Intelligence (CTI). However, due to the multi-source and heterogeneous characteristics of CTI, it is highly fragmented and requires much time for manual interpretation. Moreover, a single CTI cannot capture the whole picture of the threat. Since a single data source can only obtain part of the information segment of the object, the information from multiple data sources can perfectly and accurately reflect the general information of the object after fusion [1]. Therefore, to improve efficacy, thoroughly examine the system's security, and offer more precise decision assistance, this paper fuses data from multi-source CTI. However, since CTI resides in disparate, heterogeneous knowledge bases and the data inside is semantically heterogeneous, it is challenging to fuse the data. In this paper, we thoroughly examine cybersecurity knowledge bases and build the attack and defense

analysis of the cybersecurity ontology (ADACO) model to fuse multi-source heterogeneous data. As a result, the pertinent information can be accessed quickly and precisely. ADACO broadens the modeling dimension in cybersecurity compared to earlier ontology models and incorporates attack and defense information to handle security events in an automated or semi-automated way.

Currently, classic passive defense technologies are no longer sufficient to meet today's security requirements in the face of emerging advanced persistent threat attacks. Attack path prediction is a powerful proactive security strategy against advanced persistent threats. However, it is challenging to adjust to quick changes in the network attack and defense posture using current approaches for attack path prediction because they have poor accuracy, difficult-to-understand outputs, and cannot integrate multi-source information properly [2]. To address the above problems, this paper combines multi-source threat information based on ADACO, creates attack scenarios, and predicts attack paths utilizing the logical linkages between each assault step. The proposed threat evolution prediction algorithm (TEPA) can present the current attack scenarios and correctly anticipate the attack paths because we fully consider the actual network environment from the attacker's point of view.

Moreover, targeted defense techniques must be quickly implemented while predicting the attack path. At present, network attacks are getting more automated and intelligent. However, the deployment of defense resources and security policies on most networks remain static, making it challenging to successfully counteract today's highly intelligent attacks. As a result, to accomplish intelligent network defense, the system should automatically derive security defense tactics. Currently, game theory is the most prevalent method for research on network security defense decisions. Game theory often assumes that attackers are rational and fully informed. However, in real offensive and defensive conflicts, these assumptions are not valid, contributing to some limitations of game theory-based defense decision approaches. Therefore, to circumvent the drawbacks of employing game theory and provide more precise and intelligent recommendations for defense techniques, this study proposed the intelligent defense decision algorithm (IDDA) based on the defense technique knowledge base.

As mentioned above, to effectively fuse heterogeneous and fragmented multi-source knowledge, this work investigates multiple security knowledge bases to create the ontology model ADACO, which addresses the issue of semantic heterogeneity among knowledge. Additionally, this study suggests the algorithm TEPA based on ADACO, which can quickly predict the direction in which threats will propagate and map out the pertinent attack and defense information. The proposed algorithm IDDA offers intelligent recommendations for defensive measures. The major contributions of this paper are as follows:

1) We conduct research and analysis across multiple cybersecurity knowledge bases to integrate heterogeneous multi-source knowledge into uniformly structured and interconnected threat information. Then, we propose the attack and defense analysis of the cybersecurity ontology model, based on which we construct a cybersecurity knowledge graph to actualize the association between heterogeneous cybersecurity knowledge bases.
2) Aiming at the weak ability of the previous model to deduce and visualize the threat situation, and the inability to quickly grasp the complete picture of the threat and defense measures while predicting the attack path, we propose a threat evolution prediction algorithm to realize the association of threat information while predicting the path and enhance the visibility of the threat evolution path based on the knowledge graph.
3) In response to the inability to make defense decisions quickly and accurately for current attacks, this paper proposes an intelligent defense decision algorithm based on the defense technique knowledge base, which automatically ranks and intelligently recommends multiple defense

technologies to help security personnel quickly find the optimal defense measures to contain the spread of threats.

The workflow of this paper is shown in Fig. 1. Firstly, security knowledge is extracted from the cybersecurity knowledge bases to construct the ADACO. We simultaneously gather the topological structure and device configuration of the target network to extract the initial information. The inference rules are then loaded into the ADACO with the extracted data. Secondly, executing the TEPA, the inference results, including threat elements and their linkages, are utilized to build the threat propagation path in the knowledge graph. Finally, IDDA is used to provide a recommended list of the multiple defense techniques that have been reasoned.

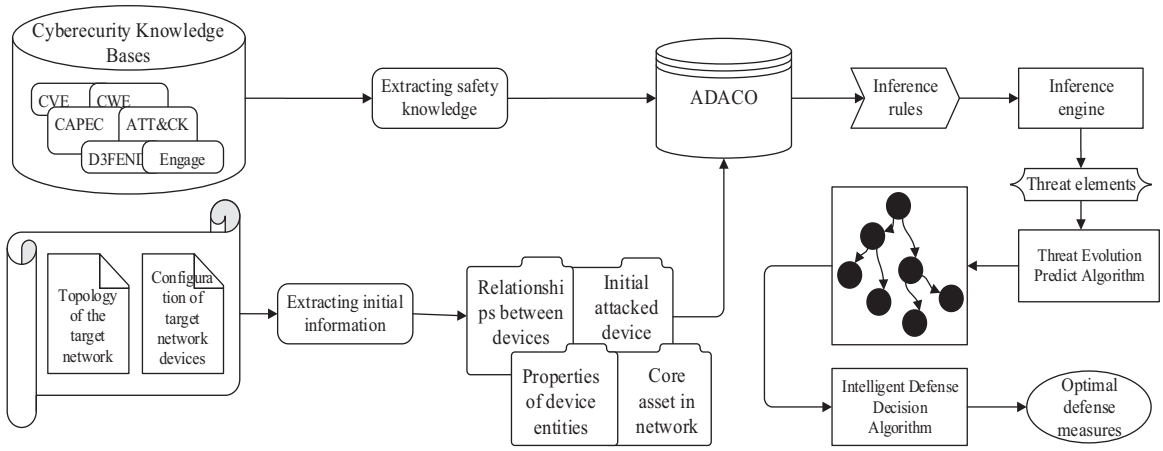

Figure 1: Workflow of the system

2 Related Work

Multi-source data fusion technology is widely used in many fields, such as the Maritime Internet of Things, remote sensing monitoring, medical diagnosis, electronic commerce, wireless communication, and fault diagnosis [3]. Liu et al. [4] developed an Augmented Reality-enabled maritime navigation system by fully integrating the visual information provided by the cameras and the positioning information provided by the automatic identification system equipment to enhance sea state awareness and vessel traffic safety. Wan et al. [5] proposed a signal-sorting method based on deep transfer learning by fusing the information data collected from multiple regions by a swarm of unmanned aerial vehicles to improve the signal-sorting accuracy of the target region. Wang et al. [6] proposed a fire station spatial layout planning method by fusing high-precision building, population space, and multi-type higher-precision Point of Interest data. Wang et al. [7] fused data from multiple sensors to provide information support for the modernization of agricultural management through environmental monitoring and automated decision-making systems. In this paper, multi-source data fusion technology is applied to network security. The current focus of multi-source data fusion applications in cybersecurity is threat modeling, where heterogeneous information in CTI is integrated into an ontology model, and correlations are extracted to analyze potential threats better and understand the cybersecurity situation to provide decision support. There have been several previous studies on ontology construction in the cybersecurity domain, as shown in Table 1.

Table 1: Ontology models in the field of cybersecurity

Ontology model	Ontology modeling object	Disadvantage
Attack analysis graphs ontology [8]	Asset, vulnerability, attack	The lack of defense techniques resulted in an inadequate definition of classes in his ontology.
CRATELO [9]	Threat, vulnerability, asset, attack, countermeasure	In CRATELO, the relationships between the classes were not sufficiently established.
Cyber ontology [10]	Time, geospatial, person, event, network operation	Because the entities remained separate, it was unable to define semantics well enough to query for entities and inter-entity relationships.
SEPSES [11]	Vulnerability, weakness, attack pattern	SEPSES was unable to create inference rules, which prevented it from mining potential data.
UCO [12]	Means, consequence, attack, attacker, attack pattern, vulnerability, exploit target	With the knowledge bases being updated continuously, UCO was unable to update itself.

To address the shortcomings of previous works, ADACO is built from multiple angles by drawing data from various knowledge bases. As a result of the data from different knowledge bases being linked, semantic heterogeneity can be removed, allowing for the construction of inference rules that helps security personnel accurately query the available knowledge and infer potential knowledge.

Our work is based on identifying attack techniques from CTI and mapping them to ADACO. However, the inevitable redundant information in CTI makes it challenging to identify attack techniques. Numerous studies and applications have been made for feature selection, which can be used to cope with redundant information. In the context of situational element gathering and fusion, Chang et al. [13] employed rough sets for attribute reduction of the original data to eliminate redundant attributes. Wu et al. [14–16] addressed the phenomenon called PCMasking and improved the accuracy and speed of Markov boundary discovery; they investigated the multi-label feature selection problem from the causal perspective and proposed the first multi-label causal feature selection algorithm; they developed a Common and Target-specific Markov boundary variable discovery (CTMB) algorithm, used it for feature selection, and proposed a novel CTMB-driven multi-label feature selection algorithm, which achieved the maximum relevance and minimum redundancy. Liu et al. [17] proposed a lightweight Internet of Things (IoT) intrusion detection model based on feature selection, using optimized machine learning methods to detect network attacks in IoT networks effectively. Usually, a CTI describing an attack event contains multiple attack techniques, and the challenge of identifying attack techniques can be handled by recasting it as a feature selection problem with multi-label classification. Therefore, to accurately identify attack techniques in CTI, this paper applies the idea of feature selection to handle redundant information.

The threat propagation path refers to a series of sequential attacks launched by an attacker to achieve his attack goal by exploiting the vulnerabilities in the target network. An important area of study in cyberspace security defense is the precise and efficient prediction of threat propagation paths [18]. Attack graphs were utilized by Chen et al. [19] to predict the attack paths. However, the coefficient values for certain crucial factors were unduly reliant on expert knowledge, making the study conclusions somewhat subjective. By simply concatenating the detected assaults, Gong et al. [20] created a threat view without considering the pre-post relationship between devices and single-step attacks, which could only predict the attack paths in straightforward scenarios. Wang et al. [2] considered the attack success probability, but the setting of attackers' capability level lacked objective calculation, so the prediction outcomes were affected. Yang et al. [21] proposed the principle of privilege promotion. However, the algorithm did not consider the impact of social engineering attacks on the threat propagation path. Although Sun et al.'s [22] threat prediction analysis method was able to anticipate the threat propagation paths with accuracy, it was unable to provide timely countermeasures. GhasemiGol et al. [23] dealt with the uncertainty of attack probability in their algorithm. Yuan et al. [24] used the breadth-first traversal algorithm in the threat path generation method. Jajodia et al. [25] constructed the topological vulnerability analysis model, which identified different attack paths starting from the initial state of the attacker. All threat paths were generated using the three algorithmic models mentioned above, which led to path redundancy. Zhang et al. [18] added a loop elimination algorithm, which could effectively avoid path redundancy and improve the efficiency of threat path generation. Still, the proposed ontology was only based on the search function of the graph database, and no inference rules were designed to explore the implicit knowledge.

The threat evolution prediction algorithm proposed in this paper takes the attacker's perspective, which considers both the probability of success in using social engineering attacks and vulnerability exploit attacks, as well as the degree of threat each device poses to core assets in the event of an attack. Additionally, it combines pre- and post-permissions to determine whether the device will likely be compromised.

3 Multi-Source Knowledge Fusion

Today, detailed information on threat events is released on different security knowledge platforms, substantially fragmenting the available information. Therefore, fragmented security knowledge needs to be fused and reconstructed to facilitate utilization.

3.1 Security Knowledge Data Sources

Among the significant cybersecurity knowledge bases, the public security knowledge bases maintained by the cybersecurity firm MITRE are widely accepted by security personnel. These knowledge bases use standardized and normalized descriptive language to represent and distribute the cybersecurity information discovered by CTI. The following are the knowledge bases consulted for this paper:

- Common Platform Enumeration (CPE) [26]
- Common Vulnerabilities and Exposures (CVE) [27]
- National Vulnerability Database (NVD) [28]
- Common Weakness Enumeration (CWE) [29]
- Common Attack Pattern Enumeration and Classification (CAPEC) [30]
- Adversarial Tactics, Techniques, and Common Knowledge Matrix (ATT&CK) [31]

- Detection, Denial, and Disruption Framework Empowering Network Defense (D3FEND) [32]
- Engage [33]

3.2 Multi-Source Knowledge Relationship Linking and Mapping

This paper implements links among the knowledge bases: CPE, CVE, NVD, CWE, CAPEC, ATT&CK, D3FEND, and Engage. Among them, the attack techniques in ATT&CK are mapped to the defense techniques in D3FEND by digital artifacts, and there is also a mapping relationship between ATT&CK and Engage. The attack patterns highlighted by CAPEC link the attack tactics and techniques in ATT&CK with the weaknesses in CWE. Weaknesses in CWE can be related to vulnerabilities in CVE, which also can be linked to NVD to view the specific description of the vulnerability entries and the Common Vulnerability Scoring System (CVSS) scores [34]. Moreover, NVD links the platforms and assets in the CPE that are affected by vulnerabilities. In conclusion, CPE, CVE, NVD, and CWE portray the affected platforms, vulnerabilities, and weaknesses the attacker exploits. CAPEC summarizes the attack patterns formed by the weaknesses that an attacker may exploit. ATT&CK provides the attack tactics and techniques. D3FEND gives defense techniques to counter these attacks. Engage offers counteracting activities to defend against these attacks actively.

From the mentioned knowledge bases, we extract multi-source cybersecurity knowledge and store the knowledge in a graph database. Specifically, the entries in each knowledge base act as nodes in the graph database, while the relational links between knowledge bases act as edges in the graph database. These edges are not bidirectional between the above knowledge bases. However, they can be traversed bidirectionally when the data is integrated into the graph structure so that data in any knowledge base can be queried by any node. Fig. 2 illustrates the relationship links among the above knowledge bases.

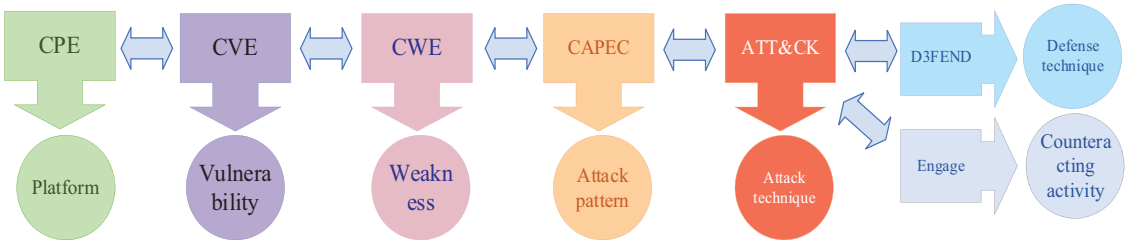

Figure 2: Schematic of relationship links

Only a few attack techniques in ATT&CK are provided with mitigations. And the rest of the attack techniques are based on the abuse of system features, making it difficult to neutralize them with preventative controls quickly. So, there are no equivalent mitigation measures in ATT&CK. Attack technique T1547.001 is shown as an example. To ultimately get higher-level privileges, the attacker causes harm by adding the malicious software to the starting folder or referencing it via the registry run key. No mitigations are offered in ATT&CK since T1547.001 is based on the abuse of system features. As seen in this example, the attack techniques of this type make security personnel cannot quickly find the mitigations they need in ATT&CK.

To address this problem, MITRE adds a brand-new "Digital Artifact" notion in the D3FEND. A digital object becomes a digital artifact when the network actor (either defensive or offensive) interacts with it in any way. The attack techniques in the ATT&CK can be linked and mapped to the defense techniques in the D3FEND thanks to the digital artifacts that operate as a bridge, which enables attack techniques based on abuse of system features to find defense techniques. For example, the corresponding digital artifacts and defense techniques in the D3FEND can be mapped by querying

the attack technique entry T1547.001. When attacked by someone using T1547.001, the attack can be countered using the relevant defense techniques in Fig. 3.

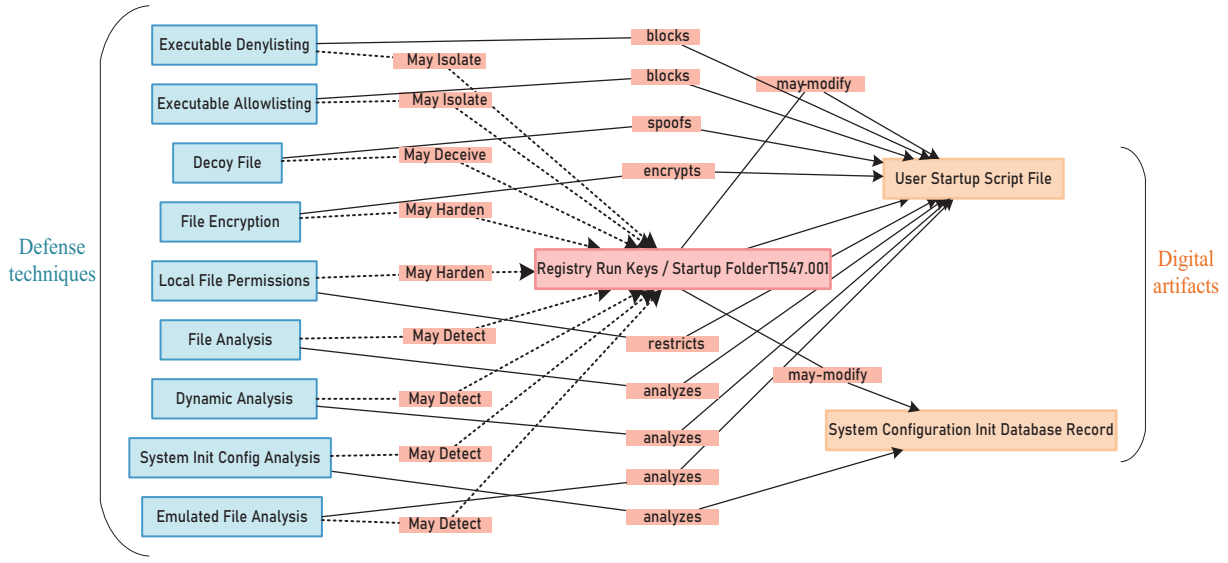

Figure 3: T1547.001 maps defense techniques in D3FEND

Using digital artifacts as a bridge, we design inference rules to make defense techniques automatically associated with attack techniques, which helps security personnel find the countermeasures quickly. Taking the attack technique T1556 as an example, the inference rules use digital artifacts as a bridge to reason out the implied relational links between the attack and defense techniques. Then, put all of them into the graph database Neo4j, as shown in Fig. 4. The red circle represents the attack technique T1556; the orange circles represent the digital artifacts associated with T1556; and the blue circles represent the defense techniques that can be used. In Fig. 4, a relationship called "hasDefend" exists between the attack technique T1556 and the defense techniques. It shows that the defense techniques can be obtained directly based on the attack techniques using inference rules without the digital artifacts.

4 Ontology Modeling and Knowledge Reasoning

Ontology can express multiple information on cyber threats as concepts with formal descriptions [20], which solves the problem that knowledge fragmentation in CTI is not conducive to expression. This paper uses a more expressive modeling language, Web Ontology Language (OWL), to construct the ontology, which provides fast and flexible data modeling capabilities and efficient automatic reasoning capabilities [35]. Then, we combine OWL ontology with the Semantic Web Rule Language (SWRL) to form inference rules [36]. We use the semantic query-enhanced web rule language (SQWRL) language to implement queries and support SWRL rules for querying and extracting knowledge after inference integration.

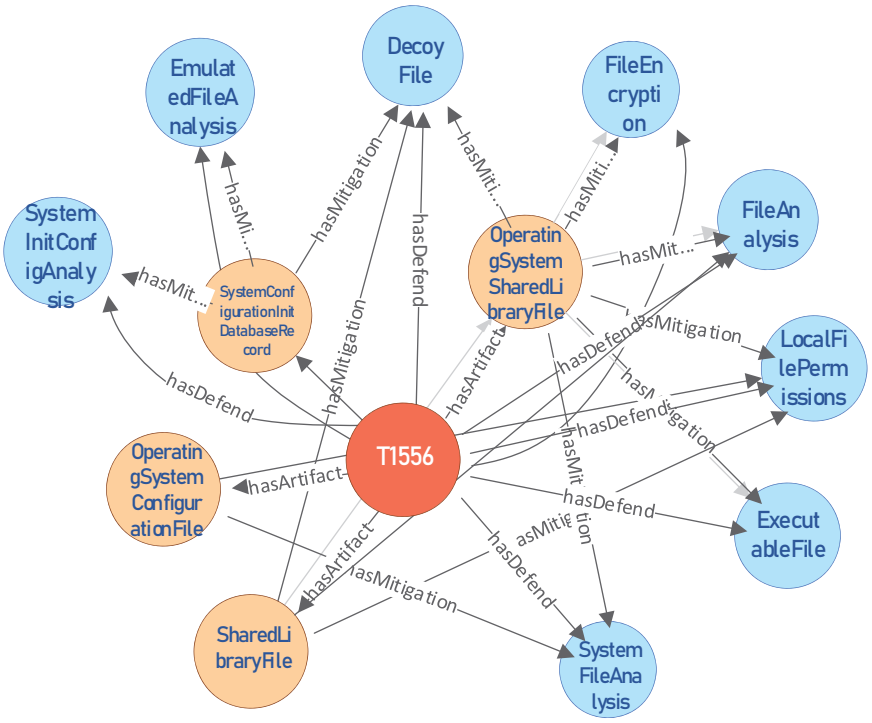

Figure 4: Mapping of attack techniques to defense techniques

In the previous sections, we effectively link multiple source knowledge bases and integrate the data from them as a source of security knowledge for building our ontology model. At the same time, we refer to several cybersecurity models and propose the attack and defense analysis of cybersecurity ontology (ADACO). We collectively refer to the data in the multi-source knowledge base, such as vulnerabilities, weaknesses, attack techniques, and defense techniques, as threat elements. ADACO uses a standard language to define classes, attributes, entities, and inter-entity relationships for threat elements to enable sharing and reuse of data. Based on ADACO, multiple inference rules are designed in this paper. The inference engine in Protégé supports the implementation of sequential multi-step inference for reasoning about the facts of threat events and underlying knowledge.

4.1 Classes and Attributes of the Ontology

ADACO contains four top-level classes, ten second-level subclasses, and several third-level subclasses. The top-level classes include: Defend, Attack, Attacker, and Device. Table 2 shows the details of the classes.

Table 2: The details of the classes

Top-level class	Second-level subclass	Attribute	Source of knowledge
Defend	D3FEND_Thing, Engage	hasArtifact, beAgainst, hasMitigation, exploit, beResisted	D3FEND, Engage

(Continued)

Table 2 (continued)

Top-level class	Second-level subclass	Attribute	Source of knowledge
Attack	ATTCK_Thing, AttackPattern, Weakness, Vulnerability	hasArtifact, beAffected, beExploited, beUsed, belong_to, hasCVSS, hasVulnerability, hasLevel, useATechnology, beResisted	ATT&CK, CAPEC, CWE, CVE
Attacker	Attacker_Name, Attacker_Vulnerability	useATechnology, beAgainst, hasAccess, hasVulnerability, hasCompromised, exploit	ATT&CK, Engage
Device	Device_Name, Asset	beAffected, hasAsset, hasAccess, hasPreRoute, hasCompromised	CPE

The inclusion relationships among classes are shown in Fig. 5a, while the logical links among the second-level subclasses are shown in Fig. 5b.

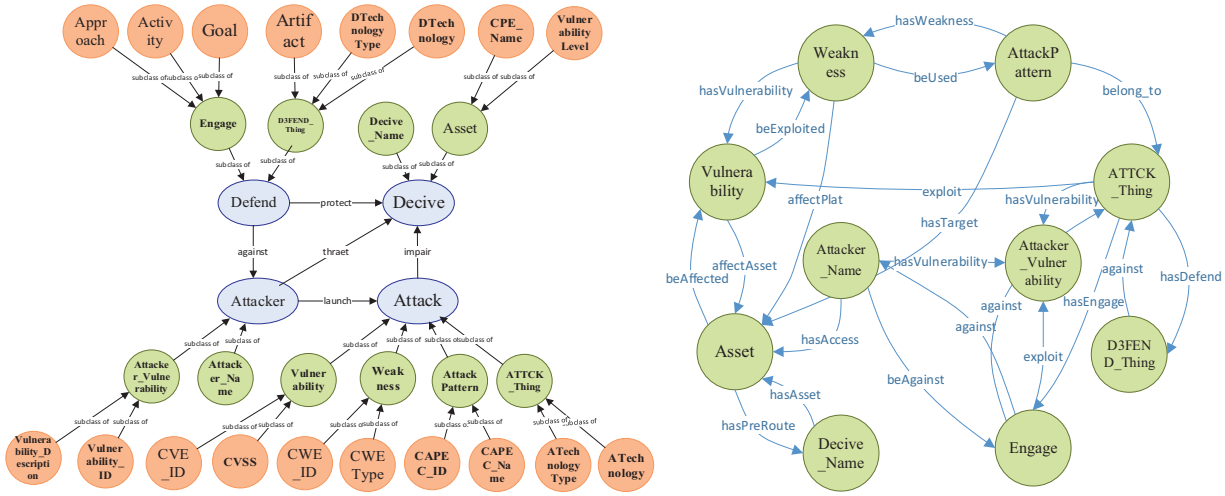

Figure 5: (a) The inclusion relationships among classes; (b) The logical links among the second-level subclasses

4.2 Design of the Inference Rules

Using inference rules allows us to reason about potential knowledge based on known knowledge, which helps us to explore new implicit relationships among threat elements. In this paper, we have designed nine inference rules to perform different functions, as shown in Tables 3 to 11.

Table 3: Inference rule 1

Subclasses	ATTCK_Thing, D3FEND_Thing
Linkages	ATT&CK → D3FEND
Inference rule	ATechnique(?t) ^ hasArtifact(?t, ?a) ^ Artifact(?a) ^ hasMitigation(?a, ?d) → hasDefend(?t, ?d)
Semantic and usage	The attack technique is associated with the affected digital artifact. Then, the digital artifact is linked to the corresponding defense technique. Ultimately, the mapping from the attack technique to the defense technique is achieved. If the security personnel know the attack technique used by the attacker, they can directly get the defense technique that corresponds to it.

Table 4: Inference rule 2

Subclasses	D3FEND_Thing
Linkages	$D3FEND_{Artifact} \rightarrow D3FEND_{DTechnique}$
Inference rule	Artifact(?f) ^ DTechnique(?d) ^ hasMitigation(?f, ?d) ^ sameAs(?f, IntranetNetworkTraffic) → sqwrl:select(?f, ?d)
Semantic and usage	The digital artifact is associated with its corresponding defense technique, different digital artifacts may be at different levels of risk, and security personnel can select the most vulnerable digital artifacts in preference to query defense techniques.

Table 5: Inference rule 3

Subclasses	Vulnerability, Asset
Linkages	CPE → CVE
Inference rule	Asset(?p) ^ beAffected(?p, ?v) ^ CVE_ID(?v) ^ hasCVSS(?v, ?s) ^ CVSS(?s) ^ hasLevel(?s, ?l) ^ VulnerableLevel(?l) → hasSeverityLevel(?p, ?l)
Semantic and usage	The asset has a vulnerability. Different vulnerabilities are classified into different severity levels, and the vulnerability level of an asset is inferred based on the severity level of the vulnerability. The severity of vulnerabilities is quantitatively assessed in the form of CVSS scores, which classify the severity of vulnerabilities into five levels: “Critical”, “High”, “Low”, “Medium”, and “None”. The inference rule automatically corresponds “Critical” and “High” to the asset’s high vulnerability level “HighLevel”; “Medium” to the asset’s medium vulnerability level “MediumLevel”; “Low” and “None” to the asset’s low vulnerability level “LowLevel”.

Table 6: Inference rule 4

Subclasses	Asset
Linkages	$CPE_{asset} \rightarrow CPE_{\text{VulnerableLevel}}$
Inference rule	Asset(?p) ^ VulnerableLevel(?l) ^ hasSeverityLevel(?p, ?l) ^ sameAs(?l,HighLevel) → sqwrl:select(?p,?l)
Semantic and usage	The asset has a vulnerability, executing the query will return the asset which has a high vulnerability level. The asset with a high vulnerability level should have a higher priority for maintenance. Security personnel can quickly query and locate the asset with a high vulnerability level to prioritize maintenance.

Table 7: Inference rule 5

Subclasses	Asset, Vulnerability, Weakness, AttackPattern, ATTCK_Thing
Linkages	CPE → CVE → CWE → CAPEC → ATT&CK
Inference rule	Asset(?p) ^ beAffected(?p, ?v) ^ Vulnerability(?v) ^ beExploited(?v,?w) ^ Weakness(?w) ^ sameAs(?w, Insufficiently Protected Credentials) ^ AttackPattern(?a) ^beUsed(?w, ?a) ^ belong_to(?a, ?t) ^ ATechnique(?t) ^ sameAs(?t, Remote Email Collection) → sqwrl:select (?p, ?t)
Semantic and usage	The asset's vulnerability is associated with a specific weakness, the weakness is associated with the attack pattern, and the attack pattern is associated with a specific attack technique, Executing the query will return the asset which has the specific weakness and is affected by the specific attack technique. Security personnel can isolate devices that have specific weaknesses and are compromised by specific attack techniques.

Table 8: Inference rule 6

Subclasses	Device, Asset, Vulnerability, Weakness, AttackPattern, ATTCK_Thing, D3FEND_Thing, Engage
Linkages	CPE → CVE → CWE → CAPEC → ATT&CK → D3FEND, Engage
Inference rule	Device(?d) ^hasAsset(?d,?p) ^Asset(?p) ^beAffected(?p,?v) ^Vulnerability(?v) ^beExploited(?v, ?w) ^ Weakness(?w) ^ beUsed(?w, ?t) ^ AttackPattern(?t) ^ belong_to(?t, ?a) ^ ATechnique(?a) ^ hasArtifact(?a, ?r) ^ Artifact(?r) ^ hasMitigation(?r, ?f) ^hasEngage(?a,?e) ^Activity(?e) → sqwrl:select(?d, ?p, ?v, ?w, ?t, ?a, ?r, ?f, ?e)
Semantic and usage	The device sequentially deduces the asset, vulnerability, weakness, attack pattern, attack technique, counteracting activity, digital artifact, and defense technique associated with it. Executing the query will return the information of all threat elements. Security personnel can query all or specific results of the inference to combat threats.

Table 9: Inference rule 7

Subclasses	Attacker_Name, Device_Name, Asset, Vulnerability
Linkages	CPE → CVE
Inference rule	Attacker(?a) ^ Device(?d) ^ Asset(?p) ^ hasAsset(?d, ?p) ^ Vulnerability(?v) ^ beAffected(?p, ?v) ^ sameAs(?v, CVE-2021-31645) ^ hasPreRoute(?p, Workstation2) → hasCompromised(?a, ?p) ^ hasAccess(?a, ?d)
Semantic and usage	An attacker attacks a device, the device owns an asset, the asset has a vulnerability and there is a device access path between the asset and the previous neighboring device. Then, it is reasoned that the attacker can compromise the device and damage the asset. At the same time, inference rule 7 can be combined with inference rule 6 to correlate the corresponding vulnerability, weakness, attack pattern, attack technique, defense measure, and confrontation activity of the compromised device promptly.

Table 10: Inference rule 8

Subclasses	ATTCK_Thing, Engage, Attacker_Vulnerability
Linkages	ATT&CK → Engage
Inference rule	ATechnique(?a) ^ hasEngage(?a, ?c) ^ Activity(?c) ^ exploit(?c, ?v) ^ Attacker_Vulnerability(?v) → hasVulnerability(?a, ?v)
Semantic and usage	The attack technique is associated with a counteracting activity that can exploit certain vulnerabilities, leading to the inference that the attack technique has those vulnerabilities. Security personnel can find out the vulnerabilities of the attack technique.

Table 11: Inference rule 9

Subclasses	ATTCK_Thing, Engage, Attacker_Name, Attacker_Vulnerability
Linkages	ATT&CK → Engage
Inference rule	Attacker_Name(?a) ^ useATechnique(?a, ?t) ^ ATechnique(?t) hasVulnerability(?t, ?v) ^ Attacker_Vulnerability(?v) ^ Activity(?c)^ exploit(?c, ?v) → hasVulnerability(?a, ?v) ^ beAgainst(?a, ?c)
Semantic and usage	The attacker uses the attack technique, which has the vulnerability. And counteracting activity exploits the vulnerability. The above information leads to reasoning about the vulnerability of the attacker and the counteracting activity that can curb the attacker. Security personnel can identify the vulnerability of the attack technique and the counteracting activity that deters the attacker.

4.3 Application of the Inference Rules

This section gives several examples of applying inference rules to demonstrate how security personnel can use them to counter threats. Two application scenarios are given below:

1) Determine the vulnerability level of the asset and whether the asset will be conquered

The email server holds the asset "arch_newsworld", which contains the vulnerability "CVE-2005-3435" with a severity level of "High". As shown in Fig. 6, the green box illustrates the security officer executing inference rule 3 to deduce that the vulnerability level of arch_newsworld is "HighLevel". At the same time, he can use the inference rule 7 to infer whether the asset will be conquered by an attacker. The red box illustrates the reasoning result that the attacker can gain complete control of the email server and compromise its asset "arch_newsworld".

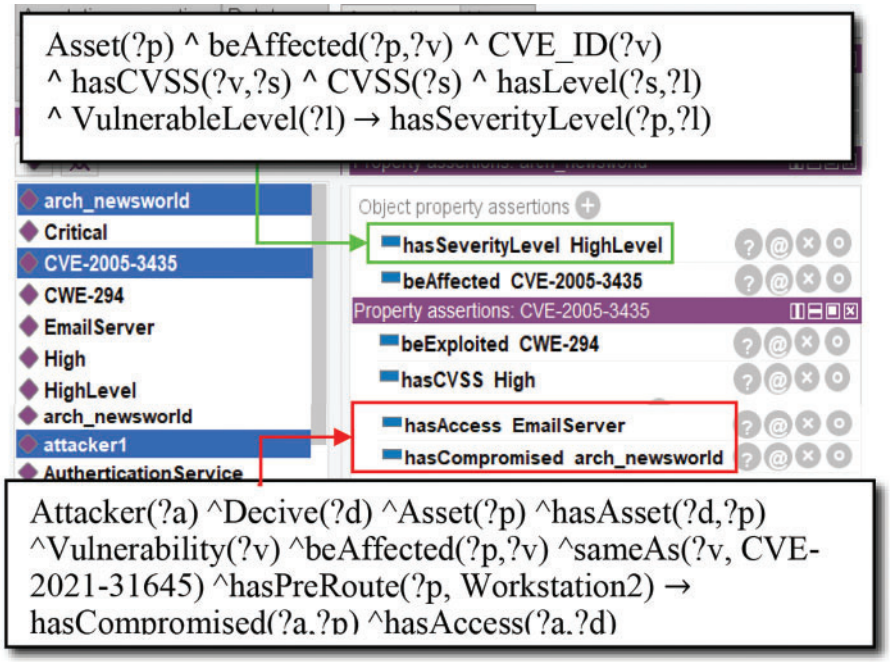

Figure 6: The result of determining the vulnerability level of the asset and whether the asset will be conquered

2) Search for information on attack and defense

When the system is under threat, the security officer can use the inference rule 6 to look up all entries of devices, assets, vulnerabilities, weaknesses, attack patterns, attack techniques, digital artifacts, defense techniques, vulnerabilities of the attack techniques, and counteracting activities. As shown in Fig. 7, the results of executing inference rule 6 are shown in the yellow box. When the system is attacked by T1114.002, the security officer can use inference rules 8 and 9 to quickly reason out the vulnerabilities of T1114.002 and the counteracting activities that can curb it. The result is shown in the red box, where the EAV entries represent the vulnerabilities of the attack technique and the EAC entries represent the counteracting activities. The security officer can also use the inference rule 1 to search for defense techniques. The green box shows the defense techniques corresponding to the two digital artifacts "ProcessCodeSegment" and "StackFrame" of T1211.

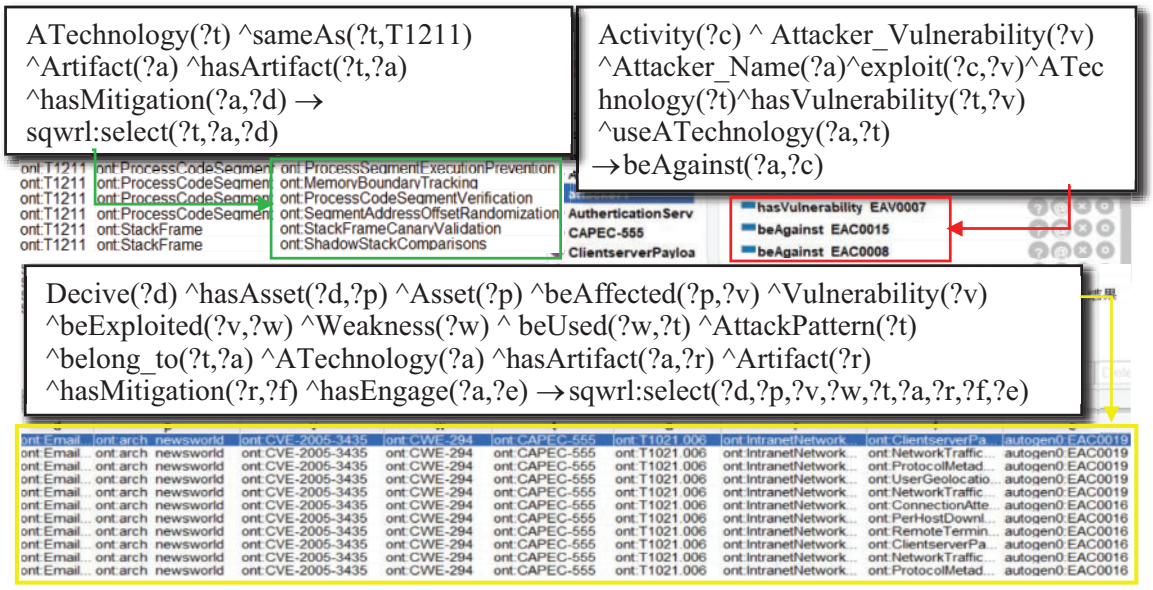

Figure 7: The result of searching for information on attack and defense

5 Main Algorithms

5.1 Meta-Path Based Threat Evolution Prediction and Correlation Response

When a threat is detected in the system, the first task is to respond to it on time and contain its spread. Therefore, it is imperative to assess and predict the development trend of the threat. The attacker will exploit vulnerabilities in the target network to launch a series of sequential attacks to achieve his attack purpose, and this set of attack sequences is defined as a meta-path by TEPA. We model the concept of "attack" as a relationship between attacker entities and device entities in a knowledge graph, thus transforming the attack prediction problem into the link prediction problem of the knowledge graph. Similarly, we link multiple threat elements into meta-paths and make all attacked device nodes connect with the corresponding threat element meta-paths. Eventually, we use the knowledge graph to show the threat evolution graph. The relevant definitions for TEPA are given below:

- Meta-path: Meta-paths are paths defined on the knowledge graph architecture layer. $P_{R_1 \circ R_2 \circ \cdots \circ R_k} = \tau(O_0) \xrightarrow{R_1} \tau(O_1) \xrightarrow{R_2} \tau(O_2) \xrightarrow{R_3} \cdots \xrightarrow{R_k} \tau(O_k)$ presents one path in knowledge graph. $\tau(O_k)$ is the entity type of O_k, and R_k denotes a type of relation. It describes a path between two entity types $\tau(O_0)$ and $\tau(O_k)$, which consists of a series of entity types $\tau(O_0) \cdots \tau(O_k)$ and a series of relation types $R_1 \cdots R_k$.
- Core asset (ASS): The asset that the attacker wants to conquer or destroy.
- Threat degree (TD): The threat degree to the core asset when the device is compromised. The higher the threat degree, the more likely the attacker selects the device for the next attack, causing the threat to propagate from this device to the core asset as a new starting point. td $\in [0, 1]$.
- Layer of topology (LOT): The layer of the device in the system topology. The higher the layer of the device, the closer it is to the core assets.
- Probability of success (POS): The probability of success of an attacker's single-step attack.

- Device set (Devices): The set includes all devices in the system.
- Business access relationship (BAR): The business access relationship between two devices d_{i-1} and d_i is represented by bar_i, i∈[1, n]. The business access relationships from device d_0 to device d_n are expressed as $d_0 \overset{bar_1}{\rightarrow} d_1 \overset{bar_2}{\rightarrow} \cdots \overset{bar_n}{\rightarrow} d_n$. Moreover, the set of business access relationships is denoted as Bar.
- Device access path (Dpath): It is an acyclic sequence of devices linked by business access relationships, i.e., the specific device d_0 is given, and the core asset is on device d_n, $dpath = \{d_0, d_1, \ldots, d_n\}$ represents the device access path from the device d_0 to d_n.
- Threat propagation path (Tpath): It is the path made up of devices the attacker can conquer with threatening means. It is an acyclic ordered sequence of interdependent single-step attacks.
- Origin device (origin): The device that the attacker first attacked.
- Pre-privilege: It is the pre-condition for an attacker to propagate a threat, i.e., a business access relationship between device d_t and the previous one d_{t-1}. The pre-privilege is extracted from the inference rule body.
- Post-privilege: It is the post-condition for an attacker to propagate a threat, i.e., an attacker launches an attack that allows him to gain complete control of device d_t. The post-privilege is extracted from the inference rule header.

Most current attack prediction algorithms ignore the importance of the attacker's psychology in the threat evolution process. Since an attacker will always choose the most advantageous means to attack the most vulnerable device, we combine the attack success probability and the threat degree of the device for prediction. The formulas for both are given separately below.

(1) Calculation of the Attack Success Probability

The attack success probability refers to the probability that an attacker will successfully conquer a device using attack means. Specifically, attack means include social engineering attacks and vulnerability exploit attacks. Social engineering attacks can be easily avoided by professional security personnel, so the attack success probability is set to 0.2. While the attack success probability of vulnerability exploit attacks is quantified based on the CVSS score.

The base score (Base) of the CVSS score reflects the inherent characteristic of vulnerability, which does not change with time and environment. The base score includes the Exploitability Subscore (ESC) and the Impact Subscore (ISC). The ESC measures the ease of vulnerability exploitation in four aspects: Attack Vector (AV), Attack Complexity (AC), Privilege Required (PR), and User Interaction (UI). The ISC measures the harm of a vulnerability in terms of confidentiality impact ($Impact_{Conf}$), integrity impact ($Impact_{Integ}$), and availability impact ($Impact_{Avail}$) [37]. The calculation formulae are shown in Eqs. (1) and (2).

$$Base = \begin{cases} Roundup\left(Min\left[(ESC + ISC), 10\right]\right), & else \\ 0, & ISC \leqq 0 \end{cases} \tag{1}$$

$$\begin{cases} \mathrm{ESC} = 8.22 * \mathrm{AV} * \mathrm{AC} * \mathrm{PR} * \mathrm{UI} \\ ISC = 1 - \left[\left(1 - Impact_{Conf}\right) \times \left(1 - Impact_{Integ}\right) \times \left(1 - Impact_{Avail}\right)\right] \end{cases} \tag{2}$$

The higher the maturity of the vulnerability code, the higher the probability that the vulnerability will be successfully exploited. So, we multiply Base by the code maturity (ExploitCodeMaturity) as the optimized score [22], multiplied by 0.1 to represent the attack success probability. The calculation formula is shown in Eq. (3).

$$pos = 0.1 * Roundup[(Base * ExploitCodeMaturity)] \tag{3}$$

(2) Calculation of the Threat Degree

When device d_0 in the device access path *dpath* is compromised, the threat degree to the core asset is calculated as follows.

i. When $dpath = \{d_0\}$, which indicates that the core asset exists in the first device of the path, and the first device has been compromised. The threat degree is calculated as Eq. (4).

$$dt_{dpath}(d_0, ass) = 1 \tag{4}$$

ii. When $dpath \neq \{d_0\}$, since the attacker propagates the threat from one device to the next by performing an attack, the threat degree of the device can be calculated only if the threat propagation path *tpath* exists on the device access path *dpath*. If the *tpath* does not exist on the *dpath*, it means that the threat cannot be propagated to the core asset along the *dpath* by attack means. As a result, the threat degree is 0.

The successful conquest of the high-topology layer device is based on the conquest of the low-topology layer device. Take the ratio of the device d_t and the core asset's topological layer numbers as the weight. The larger the value of this weight means that the device d_t is closer to the core asset. Furthermore, if the attacker wants to compromise device d_t, he must conquer all the devices on the threat propagation path before device d_t. So, this weight is then multiplied by the multiplication of the attack success probability of all devices on the threat propagation path passed from the starting device d_0 to device d_t. In this case, the threat degree is calculated as Eq. (5).

$$dt_{tpath}(d_t, ass) = \begin{cases} \frac{lot_{dt}}{lot_{ass}} * \left(\prod_{d \epsilon tpath} pos(d)\right), & tpath \neq \varnothing \\ 0, & tpath = \varnothing \end{cases} \tag{5}$$

If there are multiple adjacent devices from device d_t to core asset, and there is the threat propagation path *tpath* on the *dpath* between device d_t and each adjoining device. Then, the device with the highest threat degree among the adjacent devices is taken as the next target to attack and propagate the threat.

The core code of the algorithm proposed in this paper is as follows:

Input: Devices, Bar, ass, origin, ADACO

1)	Initialize *tpath*, Privileges, AS;	// Initialize threat propagation path "*tpath*", permission set "Privileges" and threat thing chain set "AS".
2)	target = extractAss(ass, Devices);	// Search for the device "target" where the core asset "ass" is held in the device set "Devices".
3)	prePrivileges, postPrivileges = attReason(origin, Bar, ADACO);	// The inference engine performs multi-step sequential reasoning. Extract the pre-privileges and post-privileges from the inference results and place them in the "prePrivileges" and "postPrivileges", respectively.
4)	function Iteration(origin):	// The function of the iterative attack.
5)	tpath.append(origin);	

(continued)

6)	Initialize degrees;	// Initialize the set of threat degrees for all adjacent devices at the next layer.
7)	localDevices = findLocal(origin, prePrivileges)	// Find all adjacent devices at the next layer based on the "prePrivileges".
8)	for local in localDevices:	// Find the next target device "local" in the adjacent device set "localDevices".
9)	if extractPost(local, postPrivileges) = = true:	// If the device's post-privilege can be found in the "postPrivileges".
10)	degree.append(calculateTd(local, tpath, target))	// Calculate the threat degree of all adjacent devices.
11)	end if;	
12)	else:	
13)	degree.append(0);	
14)	end else;	
15)	end for;	
16)	maxd = maxDevice(degree, localDevices)	// Extract the device with the highest threat degree among all adjacent devices to be the next target device.
17)	if maxd ! = target:	
18)	Iteration(maxd);	
19)	end if;	
20)	end;	
21)	Iteration(origin);	
22)	for device in tpath:	
23)	tpath.append(attScenario(device));	// Query the threat thing chain for all devices in *tpath*.
24)	end for;	

Output: Threat propagation path "*tpath*"

The semantics of the above algorithm is: Step 1) inputs the business access relationships and the initial attacked device. Step 2) scans all devices in the system to lock the location of the initial attacked device and the device carrying the core asset, determining the starting and ending points for constructing the device access path. Step 3) uses the inference engine to perform multi-step attack inference to obtain the sets of pre-privilege and post-privilege from the inference results. Steps 4)∼21) are the core of the algorithm. Gaining complete control of the device requires both conditions simultaneously: 1. There is a device access path between the device to be attacked and the adjacent device that is currently under complete control of the attacker. 2. The device to be attacked has a vulnerability. So, we first extract the pre-privileges to create the device access path, and then extract the post-privileges to determine whether the threat propagation path exists between devices. If the threat propagation path exists between the device and each adjacent device, the device with the highest threat degree among the adjacent devices is taken as the next target to attack. Afterward, create the directed edge to construct the complete threat propagation path. Steps 22)∼24) extract threat elements

such as vulnerabilities, weaknesses, attack techniques, defense techniques, and counteracting activities associated with each device in the propagation path and link them into a meta-path. Finally, import the output into the knowledge graph Neo4j.

5.2 Intelligent Defense Decision-Making

When the network is under attack, the system will respond to the threat and automatically return one or more defense techniques. However, security personnel may not know how to choose when faced with multiple defense techniques. In this paper, we propose the IDDA, an intelligent defense decision algorithm, to intelligently help security personnel make defense decisions.

IDDA quantitatively calculates multiple metrics from several dimensions to draw the list of recommended optimal defense techniques for the attack techniques. The quantitative metrics involved are shown below:

1) The probability of using each defense technique against the attack tactics

According to the hierarchical structure of the D3FEND, the defense tactics are divided into different defense techniques. This paper presents statistics on the number of defense techniques relevant to each attack tactic, which reflects the probability of using each defense technique against the attack tactics.

2) The number of digital artifacts covered by the defense technique

An attack technique affects one or more digital artifacts, and a defense technique acts on one or more digital artifacts, as shown in Fig. 3 in Section 3. So, the more digital artifacts involved in the defense technique, the more comprehensive the defense.

3) The similarity of the textual description of defense techniques and attack techniques

Inspired by the work of Akbar et al. [38], we use Roberta to calculate the textual similarity between the defense technique description and the attack technique description. The higher the similarity, the more the defense technique fits the attack technique.

The weights, ranking, and scores of three quantitative metrics are shown in Table 12.

Table 12: Weights, ranking, and scores of three quantitative metrics

Quantitative metric	Weight	Ranking	Score
The probability of using each defense technique against the attack tactics	0.3	1	30
		2	25
		3	20
The number of digital artifacts covered by the defense technique		4	15
		5 and later	10
The similarity of the textual description of defense techniques and attack techniques	0.4	1	40
		2	35
		3	30
		4	25
		5 and later	20

6 Experiment

To verify the validity of our work, we construct an enterprise information system as the scenario for the experiment. The instances in the scenario are mapped in the ADACO. Also, to verify the feasibility of modeling with multi-source security knowledge as an ontology, this section first gives a linking example, presented as a knowledge graph. Then, we use the TEPA for the experimental scenario to predict the threat propagation path. Finally, we use the IDDA to rank the multiple defense techniques associated with the devices in the path.

6.1 Scenario of the Experiment

The experiment scenario is shown in Fig. 8, where the system consists of four subnetworks. Subnet 1 deploys a firewall, a web server, and a file server. The file server stores critical office information of the enterprise; Subnet 2 deploys a web server, an email server, and two administration stations. The network lines of two administration stations are connected from the same router. Specifically, administration station 1 only has access to web server 2, while administration station 2 only has access to the email server; Subnet 3 deploys a workstation and a data server, and the data server stores essential business data. Workstation 1 has the user account for the file server, and access to control the data server; Subnet 4 is connected from Subnet 2. It deploys a workstation and a file transfer protocol server (FTP server). Table 13 presents the CVE entries, threat degrees, and attack types corresponding to the vulnerabilities of the devices in the system. Table 14 shows the business access relationships between the devices.

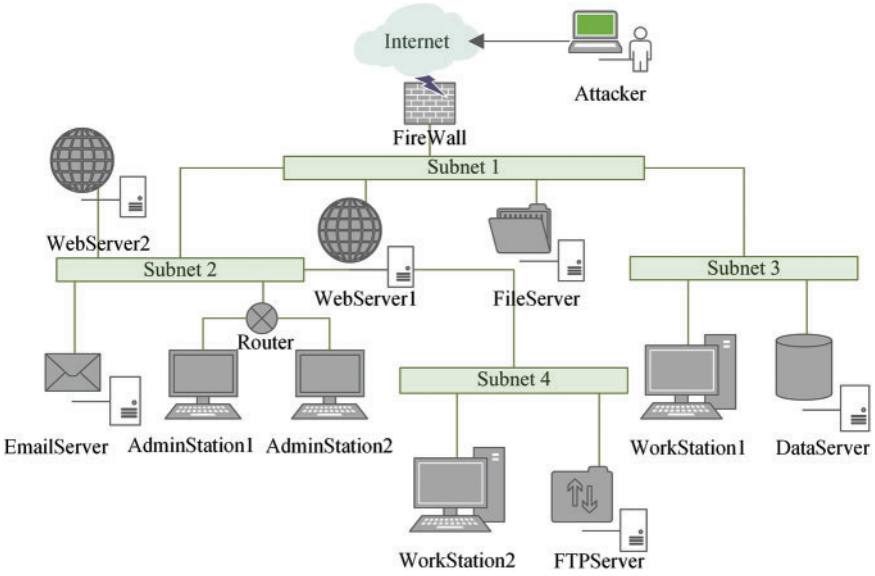

Figure 8: Topology of the scenario

6.2 Links of the Meta-Paths

This section provides an example to specify the linking relationships in the meta-paths between different classes of ADACO. The illustration shows the linkages of the device "Administration Station 2 (AS_2)" with vulnerabilities, weaknesses, attack patterns, attack techniques, digital artifacts, defense techniques, and counteracting activities. The entities in the example are listed in Table 15.

Table 13: Instances and their properties

Device	Asset	Vulnerability	CVSS	Attack Type
Web server 1	cloud_foundation	CVE-2021-21972	9.8	Privilege escalation
Web server 2	mac_os_x	CVE-2014-1266	5.8	Privilege escalation
Workstation 1	phpBB	CVE-2005-0603	5.0	Discovery
Workstation 2	matrix_screen_saver	CVE-1999-1454	4.6	Defense evasion
Router	rv_110w	CVE-2022-20923	9.8	Defense evasion
Firewall	wordfence_security	CVE-2022-3144	4.8	Script injection
Data server	SQL	CVE-2004-0366	7.5	SQL injection
FTP server	glFTPd	CVE-2021-31645	7.5	Excessive allocation
Admin station 1	google_chrome	CVE-2018-6116	6.5	Code execution
Admin station 2	extcalendar	CVE-2007-0681	7.5	Privilege escalation
Email server	arch_newsworld	CVE-2005-3435	7.5	Defense evasion
File server	linux	CVE-2009-1630	4.4	Defense evasion

Table 14: Business access relationships

From	To	Object properties
Web server 1	Router (rv_110w)	hasRoute (Web_1, rv_110w)
Workstation 1	Data server (SQL)	hasRoute (Work_1, SQL)
Workstation 2	FTP server (glFTPd)	hasRoute (Work_2, glFTPd)
Firewall	Web server 1 (cloud_foundation)	hasRoute (FW, cloud_foundation)
Firewall	File server (linux)	hasRoute (FW, linux)
File server	Workstation 1 (phpBB)	hasRoute (FS, phpBB)
Router	Admin station 1 (google_chrome)	hasRoute (Router, google_chrome)
Router	Admin station 2 (extcalendar)	hasRoute (Router, extcalendar)
Admin station 1	Web server 2 (mac_os_x)	hasRoute (AS_1, mac_os_x)
Admin station 2	Email server (arch_newsworld)	hasRoute (AS_2, arch_newsworld)
Email server	Workstation 2 (matrix_screen_saver)	hasRoute (ES, matrix_screen_saver)

Inject AS_2 into the inference engine and import the inference results into the knowledge graph. The linkages are shown in Fig. 9. And Table 16 illustrates the linkages between the above entities in the form of meta-paths.

Table 15: The entities in the example

Asset	extCalendar	Vulnerability	CVE-2007-0681
Severity level	High	Weakness	CWE-522
Vulnerability level	HighLevel	Attack pattern	CAPEC-555
Attack technique	T1021.006, T1114.002		
Digital artifact	Intranet network traffic, mail server		
Defense technique	Decoy network resource, remote terminal session detection, per host download upload ratio analysis, network traffic community deviation, connection attempt analysis, client-server payload profiling, user geolocation logon pattern analysis, network traffic filtering		
Vulnerability of the attack technique	EAV0001, EAV0002, EAV0007, EAV0010, EAV0019, EAV0020		
Counteracting activity	EAC0002, EAC0006, EAC0016, EAC0019		

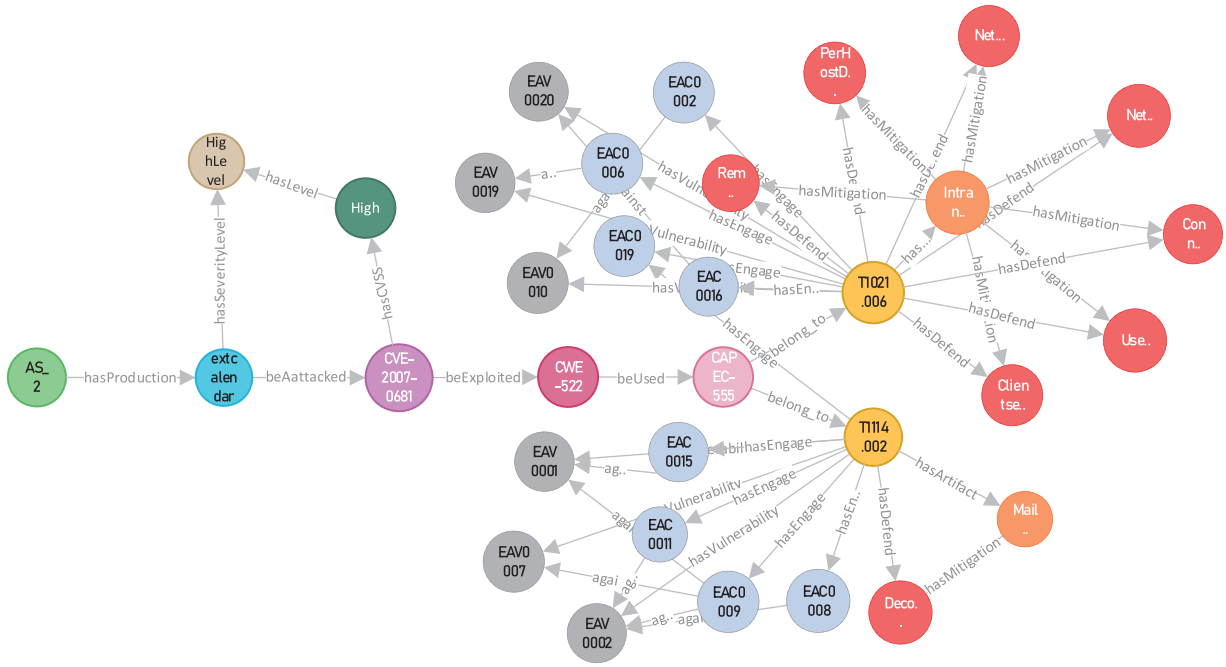

Figure 9: The linkages in the example

Table 16: Linkages in the meta-paths

P_1: Device → Asset → Vulnerability → Severity level → Vulnerability level
P_2: Device → Asset → Vulnerability → Weakness → Attack pattern → Attack technique → Counteracting activity → Vulnerability of the attack technique
P_3: Device → Asset → Vulnerability → Weakness → Attack pattern → Attack technique → Digital artefact → Defense technique

6.3 Prediction of the Threat Evolution

This section verifies the validity of TEPA based on the scenario in Section 6.1. The initial conditions for the experiment are given below:

(1) The attacker attacked the firewall, which suffered from a malicious script injection vulnerability "CVE-2022-3144".
(2) The firewall holds the business access relationships between both Web Server 1 and File Server.
(3) The core asset is hosted on the FTP Server.

Execute the TEPA to predict the devices most likely to be compromised by each attack step and then link them as the meta-path. At the same time, the threat elements associated with these devices are also linked to the path. The final results are imported into Knowledge Graph, as shown in Fig. 10. We mark the threat propagation path with black arrows. Based on the predicted path, security personnel can quickly get a picture of threats from the knowledge graph and take appropriate defensive measures for each attack step to contain the spread of the threat. We have compiled the results of our experiment and presented them in Table 17. For simplicity of expression, the devices in Table 17 are replaced by abbreviations, e.g., the firewall is written as FW.

Based on the table, we analyze the attacker's actions in this experiment as follows:

(1) The attacker attacked Firewall, which owned the software "Wordfence_Security". And the vulnerability in Wordfence_Security, CVE-2022-3144, caused it to under-translate stored values, which allowed the attacker to inject malicious web scripts into the settings. When the user visited the page affected by the setting, it was subject to cross-site scripting by the attacker, resulting in the firewall being completely compromised.
(2) The attacker then attacked Web Server 1 (Web_1), which owned the software "cloud_foundation". The cloud_foundation contained a remote code execution vulnerability, CVE-2021-21972, which allowed the attacker to execute commands with unrestricted privileges and thus gain complete control of Web_1.
(3) There was a business access path between Web_1 and Router. The Router was configured with hardware "rv_110w". The rv_110w had the vulnerability "CVE-2022-20923", which allowed the unauthenticated attacker to bypass authentication and access the network.
(4) The attacker attacked Administration Station 2 (AS_2) along the network. AS_2 held the software "extCalendar", which had the vulnerability "CVE-2007-0681". CVE-2007-0681 allowed the attacker to steal the user's password and gain complete control of AS_2.
(5) The attacker accessed the mail server (ES) via AS_2. ES contained the software "arch_newsworld", which suffered from the vulnerability "CVE-2005-3435". The attacker exploited CVE-2005-3435 to obtain the hash of the user's password to bypass authentication and gain complete control of the ES.

(6) The attacker accessed Workstation 2 (Work_2) via ES. Work_2 held the screensaver "matrix_screen_saver", which had the elevation of privilege vulnerability "CVE-1999-1454". It allowed the attacker to bypass the password prompt by pressing the ESC key and gain complete control of Work_2.

(7) Via Work_2, the attacker could access FTP Server. The software "glFTPd" in FTP Server had the vulnerability "CVE-2021-31645", which could enable the attacker to cause a threat event of denial service by exceeding the connection limit.

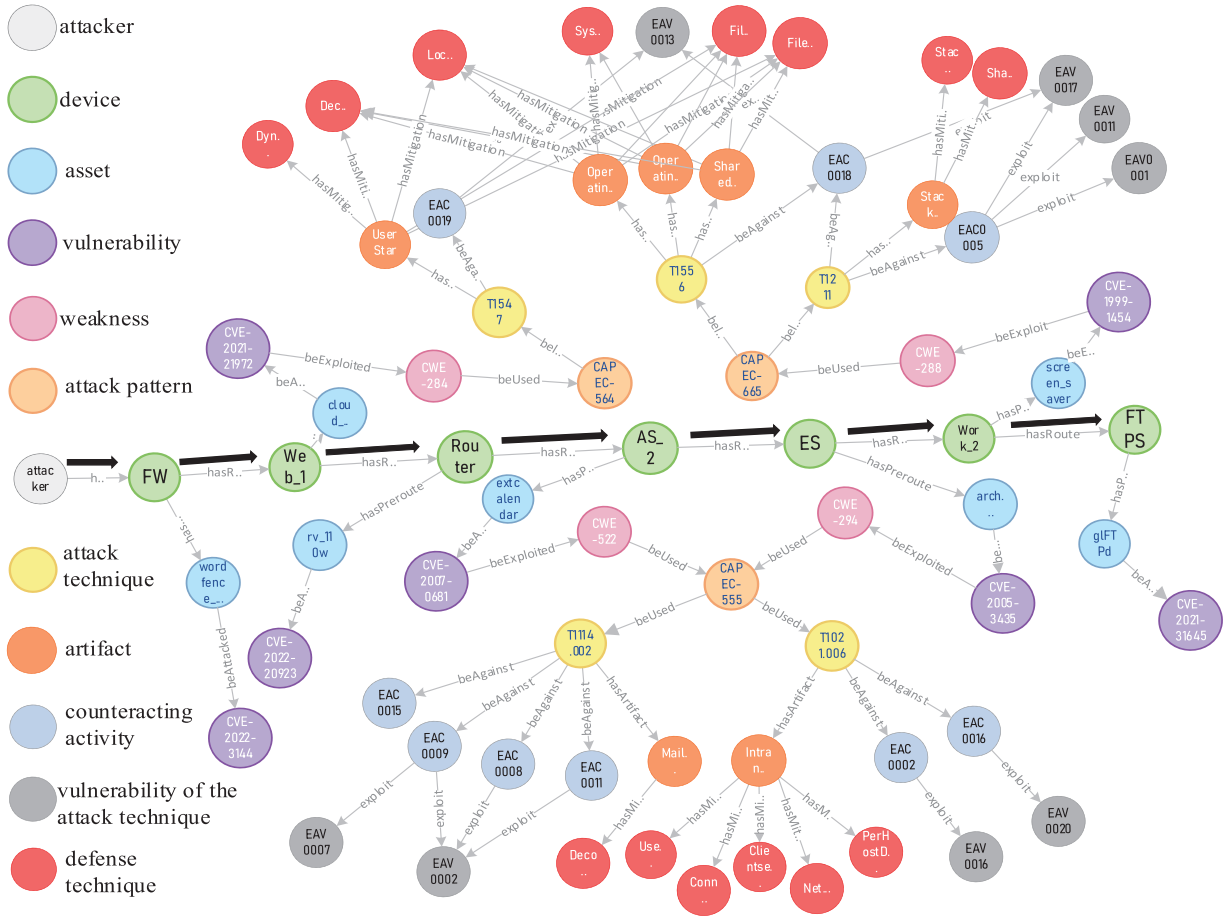

Figure 10: The result of the threat propagation path

Through the above analysis, we can observe the attack steps and verify the validity of the proposed methodological model in this paper. ADACO integrates the "defense" to extend the modeling knowledge of the cybersecurity domain. Table 18 compares ADACO with other ontology models.

We compare TEPA with previous algorithms in Table 19. The comparison result shows that TEPA is relatively improved, which takes the threat impact elements (probability of the successful attack, threat degree) into account and avoids the problem of path redundancy. Moreover, only the work in this paper can predict the threat propagation path while correlating the attacked devices in the path with their threat elements, such as vulnerabilities, weaknesses, attack techniques, and defense techniques, which enriches the prediction results.

Table 17: The compiled information on the threat propagation path

Device	Topology layer	Threat degree	Be the next target device	The next target device	The prediction of the threat propagation path
FW	1	0.0686	—	Web_1	FW→Web_1
Web_1	2	0.1344	Yes	Router	FW→Web_1→Router
FS	2	0.0603	No	—	—
Work_1	3	0.0453	No	—	—
DS	4	0.0453	No	—	—
Router	3	0.1976	Yes	AS_2	FW→Web_1→Router→AS_2
AS_1	4	0.1712	No	—	—
Web_2	5	0.1241	No	—	—
AS_2	4	0.1976	Yes	ES	FW→Web_1→Router→AS_2 →ES
ES	5	0.1852	Yes	Work_2	FW→Web_1→Router→AS_2 →ES→Work_2
Work_2	6	0.1022	Yes	FTPS	FW→Web_1→Router→AS_2 →ES→Work_2→FTPS

Table 18: Comparison among the cybersecurity ontology models

Ontology	Asset	Vulnerability	Weakness	Attack pattern	Attack technique	Defense technique	Support inference
Wu et al. [8]	✓	✓	×	×	✓	×	✓
CRATELO [9]	×	✓	×	×	✓	✓	✓
Kiesling et al. [11]	✓	✓	✓	✓	×	×	×
UCO [12]	✓	✓	×	✓	✓	×	✓
Zhang et al. [18]	×	✓	×	×	×	×	×
Yang et al. [21]	✓	✓	×	×	×	×	×
Sun et al. [22]	✓	✓	×	×	✓	×	✓
Yuan et al. [24]	×	✓	×	×	×	×	×
TVA [25]	×	✓	×	×	✓	×	×
ADACO	✓	✓	✓	✓	✓	✓	✓

6.4 Implementation of Intelligent Defense Decision-Making

6.4.1 Contrast Analysis

TEPA has associated with several defense techniques in Section 6.3. And in this section, we use the IDDA to rank and recommend them. The recommendation result is compared with the algorithm of Akbar et al. [38] to demonstrate IDDA's superiority. Akbar et al. [38] analyzed textual descriptions of attack techniques and defense techniques using the standard model "Roberta". Roberta offers deep semantic knowledge to derive meaningful associations between attack techniques and defense

techniques. They matched attack techniques to defense techniques and provided a ranked list of defense techniques for each attack technique.

Table 19: Comparison among the threat propagation path prediction algorithms

Algorithm	No redundant paths	Take threat impact elements into account	Correlate the threat elements
Wang et al. [2]	×	✓	×
Wu et al. [8]	×	×	×
Zhang et al. [18]	×	×	×
Chen et al. [19]	✓	×	×
Yang et al. [21]	×	✓	×
Sun et al. [22]	✓	✓	×
GhasemiGol et al. [23]	×	✓	×
Yuan et al. [24]	×	×	×
TVA [25]	×	×	×
TEPA	✓	✓	✓

Taking the attack technique "T1547" as an example, IDDA and the algorithm of Akbar et al. are executed, and the ranking results are given separately, as shown in Table 20.

Table 20: The ranking of defense techniques

Attack technique	Ranking of defense techniques	Ranking of IDDA	Ranking of Akbar et al.'s algorithm [38]
T1547	Local file permissions	1	3
	File encryption	2	5
	File analysis	3	4
	Executable allowlisting	4	2
	Dynamic analysis	5	1
	Emulated file analysis	6	6
	Decoy file	7	8
	Asset vulnerability	8	11
	Executable denylisting	9	9
	Configuration inventory	10	10
	System init config analysis	11	7

The analysis of the ranking results of both shows that IDDA gives approximately the same result as Akbar et al.'s algorithm [38], which proves the effectiveness of IDDA. To further validate the reasonableness and accuracy of our ranking result, we analyze the above eleven defense techniques using expert knowledge and select the four most effective ones. In the list ranked by the IDDA, the four most effective defense techniques selected overlap three of the top four in the list, while using the

algorithm of Akbar et al. [38], the four most effective defense techniques selected only overlap two of the top four, which shows that the IDDA is more accurate and reliable.

6.4.2 Statistical Analysis

The three indicators in Table 21 are used simultaneously by IDDA for calculation, while indicator 1 is the sole indication used by Akbar's algorithm. The outcomes are contrasted for the four cases in Table 21 to demonstrate the effectiveness of the remaining indicators used in IDDA:

Table 21: Indicators of recommendation and the four cases

Indicator 1	The similarity of the textual description of defense techniques and attack techniques
Indicator 2	The probability of using each defense technique against the attack tactics
Indicator 3	The number of digital artifacts covered by the defense technique
Case1	indicator 1 (Akbar's algorithm)
Case2	indicator 1 + indicator 2
Case3	indicator 1 + indicator 3
Case4	indicator 1 + indicator 2 + indicator 3 (IDDA).

We randomly select one hundred attack techniques from the ATT&CK matrix and recommend the relevant defense techniques in the above four cases, obtaining four hundred sets of ranking results of defense techniques in total. In keeping with the principle of the prior experiment, we use expert knowledge to examine all defense techniques in each group and select the four most effective techniques. The top four in the ranked list are compared with the four most effective defense techniques to calculate the number of overlaps between the top four and the most effective defense technique. We counted the number of overlaps in these four cases and calculated their probabilities. The comparison results are shown in Fig. 11.

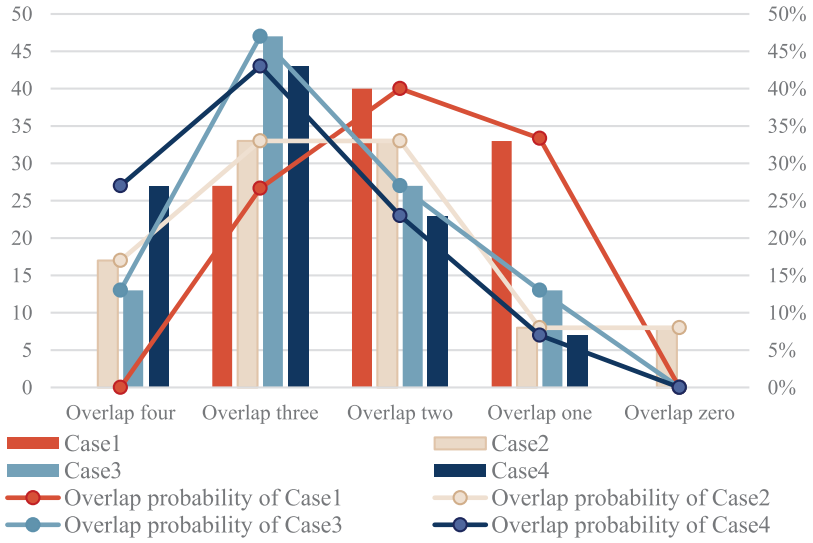

Figure 11: Number of overlaps and overlap probability

According to Fig. 11, Case 4 has the best overall performance since it has the highest probability of overlapping four, and the total probability of overlapping four and overlapping three is the highest, although the probability of overlapping three is just slightly lower than Case 3. Case 1 performed poorly, with the lowest probability of overlapping four and overlapping three as well as the highest probability of overlapping two and overlapping one. Case 3's curve trend resembles Case 4, whereas Case 2's curve trend falls between Case 1 and Case 4. The probability of overlapping three decreases when indicator 2 is added to Case 3, indicating that indicator 3 has a more favorable impact on the calculation. Even though adding indication 2 somewhat reduces the probability of overlapping three, it increases the overall overlapping probability. So far, the effectiveness and superiority of IDDA have been confirmed by the above evaluations.

7 Conclusion

In this work, we employ several cybersecurity knowledge bases as sources of information, integrate multi-source information on items like the asset, vulnerability, weakness, attack pattern, attack technique, defense technique, and counteracting activity, and organize relationships between them. Based on this, we build the ADACO model and map it to the knowledge graph, resolving the semantic heterogeneity issue and laying the foundation for knowledge retrieval. Nine inference rules that may be used in an actual Internet situation have been developed for ADACO. Additionally, when the system is under attack, ADACO combines the TEPA for predicting the threat propagation path and links threat information to each compromised device. Finally, the IDDA gives security professionals a practical means of making the most effective decisions in the case of an assault.

For future work, firstly, the threat propagation path algorithm does not consider the case where multiple post-permissions must be satisfied simultaneously to compromise a particular device. So, we will further refine the classification of the types of vulnerabilities and the required permissions to fill the gap. Secondly, we will provide ADACO with more threat elements on malicious families to automate attack attribution. Finally, the experiment shows that the indicator "the number of digital artifacts covered by the defense technique" plays a more positive role in the IDDA. As a result, we will try to increase the weight of this indicator to improve the overall probability of overlapping, thereby improving the accuracy of the defense technique recommendation.

Acknowledgement: The authors would like to thank all those who have contributed in this area and the anonymous reviewers for their valuable comments and suggestions, which have improved the presentation of this paper.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Shuqin Zhang, Xinyu Su; data collection: Yunfei Han; analysis and interpretation of results: Peiyu Shi, Tianhui Du; draft manuscript preparation: Xinyu Su. The authors declare that they have no conflicts of interest to report regarding the present study.

Availability of Data and Materials: The ontology involved in this paper can be obtained by sending an E-mail to the corresponding author.

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

References

[1] P. F. Zhang, T. R. Li, G. Q. Wang, D. X. Wang, P. Lai *et al.,* "A multi-source information fusion model for outlier detection," *Information Fusion*, vol. 93, pp. 192–208, 2023.

[2] S. Wang, G. M. Tang and G. Kou, "Attack path prediction method based on causal knowledge net," *Journal on Communications*, vol. 37, no. 10, pp. 188–198, 2016.

[3] P. F. Zhang, T. R. Li, G. Q. Wang, C. Luo, H. M. Chen *et al.,* "Multi-source information fusion based on rough set theory: A review," *Information Fusion*, vol. 68, pp. 85–117, 2021.

[4] R. W. Liu, Y. Guo, J. T. Nie, Q. Hu, Z. H. Xiong *et al.,* "Intelligent edge-enabled efficient multi-source data fusion for autonomous surface vehicles in maritime internet of things," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1574–1587, 2022.

[5] L. T. Wan, R. Liu, L. Sun, H. S. Nie and X. P. Wang, "UAV swarm based radar signal sorting via multi-source data fusion: A deep transfer learning framework," *Information Fusion*, vol. 78, pp. 90–101, 2021.

[6] S. H. Wang, W. Hu and M. Su, "Research on new urban fire station location based on multi-source data integration," *Geospatial Information*, vol. 21, no. 5, pp. 63–66, 2023.

[7] O. W. Wang, L. Wang and J. H. Fu, "Huawei Hongmeng smart agriculture system based on multi-source data fusion algorithm," *Journal of Smart Agriculture*, vol. 3, no. 9, pp. 1–4, 2023.

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

[9] A. Oltramari, L. F. Cranor, R. J. Walls and P. McDaniel, "Building an ontology of cyber security," *CEUR Workshop Proceedings*, vol. 1034, pp. 54–61, 2014.

[10] M. Iannacone, S. Bohn, G. Nakamura, J. Gerth, K. Huffer *et al.,* "Developing an ontology for cyber security knowledge graphs," in *Proc. of the 10th Annual Cyber and Information Security Research Conf. Association for Computing Machinery*, New York, NY, USA, pp. 1–4, 2015.

[11] E. Kiesling, A. Ekelhart, K. Kurniawan and J. Fajar, "The SEPSES knowledge graph: An integrated resource for cybersecurity," *The Semantic Web–ISWC 2019*, vol. 11779, pp. 198–214, 2019.

[12] Z. Syed, A. Padia and T. Finin, "UCO: A unified cybersecurity ontology," in *Workshops at the Thirtieth AAAI Conf. on Artificial Intelligence*, Vancouver, British Columbia, Canada, pp. 195–202, 2016.

[13] Y. H. Chang, Z. R. Ma and X. Li, "Extraction of security situation elements based on probabilistic neural network," *Cyberspace Security*, vol. 11, no. 10, pp. 56–61, 2020.

[14] X. Wu, B. B. Jiang, K. Yu, C. Miao and H. Chen, "Accurate Markov boundary discovery for causal feature selection," *IEEE Transactions on Cybernetics*, vol. 50, no. 12, pp. 4983–4996, 2020.

[15] X. Y. Wu, B. B. Jiang, K. Yu, H. H. Chen and C. Y. Miao, "Multi-label causal feature selection," in *The 34th AAAI Conf. on Artificial Intelligence (AAAI'20)*, New York, NY, USA, pp. 6430–6437, 2020.

[16] X. Y. Wu, B. B. Jiang, Y. Zhong and H. H. Chen, "Multi-target Markov boundary discovery: Theory, algorithm, and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4964–4980, 2023.

[17] X. Y. Liu, T. L. Lu and Y. H. Du, "Lightweight IoT intrusion detection method based on feature selection," *Netinfo Security*, vol. 23, no. 1, pp. 66–72, 2023.

[18] K. Zhang and J. J. Liu, "A threat path generation method based on knowledge graph," *Computer Simulation*, vol. 39, no. 4, pp. 350–356, 2022.

[19] R. Y. Chen, Z. M. Chen and H. Wang, "Research on threat modeling of industrial control network based on attack graph," *Netinfo Security*, vol. 18, no. 10, pp. 70–77, 2018.

[20] L. Gong, R. B. Si and Y. Tian, "Research on key technologies of ontology based threat modeling for cyber range," *Journal of CAEIT*, vol. 15, no. 12, pp. 1139–1144, 2020.

[21] Y. J. Yang, Q. Leng, R. X. Pan and H. Hu, "Research on dynamic threat tracking and quantitative analysis technology based on attribute attack graph," *Journal of Electronics & Information Technology*, vol. 41, no. 9, pp. 2172–2179, 2019.

[22] C. Sun, H. Hu, Y. J. Yang and H. Q. Zhang, "Two-layer threat analysis model integrating macro and micro," *Chinese Journal of Network and Information Security*, vol. 7, no. 1, pp. 143–156, 2021.

[23] M. GhasemiGol, A. Ghaemi-Bafghi and H. Takabi, "A comprehensive approach for network attack forecasting," *Computers & Security*, vol. 58, pp. 83–105, 2016.
[24] B. T. Yuan, Z. L. Pan, F. Shi and Z. H. Li, "An attack path generation methods based on graph database," in *IEEE 4th Information Technology, Networking, Electronic and Automation Control Conf. (ITNEC)*, Chongqing, China, pp. 1905–1910, 2020.
[25] S. Jajodia, S. Noel and B. O'Berry, "Topological analysis of network attack vulnerability," *Managing Cyber Threats*, vol. 5, pp. 247–266, 2005.
[26] NIST. Common Platform Enumeration. [Online]. Available: https://nvd.nist.gov/Products/CPE
[27] MITRE. Common Vulnerabilities and Exposure. [Online]. Available: https://cve.mitre.org/
[28] NIST. National Vulnerability Databased. [Online]. Available: https://nvd.nist.gov
[29] MITRE. Common Weakness Enumeration. [Online]. Available: https://cwe.mitre.org/
[30] MITRE. Common Attack Pattern Enumeration and Classification. [Online]. Available: https://capec.mitre.org/
[31] MITRE. ATT&CK Matrix for Enterprise. [Online]. Available: https://attack.mitre.org/
[32] MITRE. D3FEND. [Online]. Available: https://d3fend.mitre.org/
[33] MITRE Engage™, *An Adversary Engagement Framework from MITRE*. https://engage.mitre.org/
[34] FIRST. Common Vulnerability Scoring System. [Online]. Available: https://www.first.org/cvss/
[35] F. Gao, H. Xiong and J. G. Gu, "Real-time semantic data stream reasoning based on knowledge representation learning," *Computer Applications and Software*, vol. 39, no. 2, pp. 26–31, 2022.
[36] F. Wang, Y. Z. Zhang and X. F. Luo, "Semantic query of ontology knowledge base based on SQWRL," *Computer Technology and Development*, vol. 27, no. 2, pp. 15–19, 2017.
[37] X. Zhang, S. G. Huang, Y. Xia and S. H. Song, "Attack graph-based method for vulnerability risk evalution," *Application Research of Computers*, vol. 27, no. 1, pp. 278–280, 2010.
[38] K. A. Akbar, S. M. Halim, Y. Hu, A. Singhal, L. Khan *et al.,* "Knowledge mining in cybersecurity: From attack to defense," *Data and Applications Security and Privacy XXXVI*, pp. 110–122, 2022.