Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.044526

ARTICLE

Application Research on Two-Layer Threat Prediction Model Based on Event Graph

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

School of Computer Science, Zhongyuan University of Technology, Zhengzhou, HEN037, China

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

Received: 01 August 2023 Accepted: 19 October 2023 Published: 26 December 2023

ABSTRACT

Advanced Persistent Threat (APT) is now the most common network assault. However, the existing threat analysis models cannot simultaneously predict the macro-development trend and micro-propagation path of APT attacks. They cannot provide rapid and accurate early warning and decision responses to the present system state because they are inadequate at deducing the risk evolution rules of network threats. To address the above problems, firstly, this paper constructs the multi-source threat element analysis ontology (MTEAO) by integrating multi-source network security knowledge bases. Subsequently, based on MTEAO, we propose a two-layer threat prediction model (TL-TPM) that combines the knowledge graph and the event graph. The macro-layer of TL-TPM is based on the knowledge graph to derive the propagation path of threats among devices and to correlate threat elements for threat warning and decision-making; The micro-layer ingeniously maps the attack graph onto the event graph and derives the evolution path of attack techniques based on the event graph to improve the explainability of the evolution of threat events. The experiment's results demonstrate that TL-TPM can completely depict the threat development trend, and the early warning results are more precise and scientific, offering knowledge and guidance for active defense.

KEYWORDS

Knowledge graph; multi-source data fusion; network security; threat modeling; event graph; absorbing Markov chain; threat propagation path

1 Introduction

Network attacks have caused irreparable economic losses to countries, companies, and individuals. One of the most effective ways of dealing with cyber-attacks today is using cyber threat intelligence (CTI). However, many CTIs are not categorized by domain, weakening the sharing effectiveness [1]. Moreover, the heterogeneity of the indicator of compromise (IOC) in CTI leads to severe fragmentation of security information, which requires much time and effort to decipher the potential relationships between them manually [2]. However, threat modeling enables the heterogeneous information in CTI to be combined into a model to understand the cyber security situation better and to provide supporting information for decision-making. At present, there has been a lot of research into threat modeling. Xu et al. [3] modeled the review dataset as a reviewer projection graph to detect opinion

spammer groups, who conducted malicious reviews aimed at misleading consumers. Zhao et al. [4] modeled and analyzed the interdependencies between heterogeneous IOCs as well as the interactions between different types of web objects in multi-source data. Their models could describe threat events more comprehensively and effectively, capture the intrinsic interactions between cyber objects and learn the evolutionary patterns of cyber threats. In addition, there are numerous threat modeling researches based on ontology, which construct ontology models specific to the cybersecurity domain. Ontology models can describe a wide range of information about cyber threats in concepts [5], solving the problem that data from different security platforms can be challenging to understand and utilize due to semantic heterogeneity. Wu et al. [6] created a security knowledge ontology that used a standard language to represent assets, vulnerabilities, and attacks. However, the ontology did not include defensive tactics, which resulted in an inadequate definition of the ontology's classes. Iannacone et al. [7] created an advanced ontology based on malware and the diamond model. Still, the structure was unclear, and entities in multiple datasets remained isolated, making it impossible to search or query for entities and inter-entity relationships. Syed et al. [8] developed the unified cybersecurity ontology, characterized and articulated using the cybersecurity standard. However, the instance data in this ontology model was inadequate and could not keep up with the knowledge base's continual upgrading. After summarizing the advantages and shortcomings of the previous work, the multi-source threat element analysis ontology (MTEAO) in this paper is built from numerous aspects utilizing data from various knowledge bases. The information in disparate knowledge bases can be linked to minimizing semantic heterogeneity, allowing inference rules to be formed to accomplish correct queries and prospective knowledge inference. At the same time, MTEAO can be regularly updated and enhanced by acquiring threat information from the outside world.

Simultaneously, APT has moved into the mainstream of today's network assaults. Traditional passive defenses are no longer enough to meet today's security requirements. Active defense can be targeted by learning and analyzing the attacker's attack preference [9]. In addition, attack path prediction is a proactive defense approach against APT assault, and graph structures are increasingly being applied to it by scholars. Knowledge graph maps the real world to the data world, which describes concepts, entities, events, and their relationships in the objective world. Based on threat modeling, the concept "attack" is described in the knowledge graph as a relation link between the attackers and devices, changing the attack path prediction issue into the link prediction issue in the knowledge graph. As a result, how to forecast the attack path correctly and effectively is an essential research topic in cyberspace defense. Currently, previous research on attack paths is divided into two main layers: the macro-layer and the micro-layer.

At the macro-layer: Hu et al. [10] proposed a multi-step attack path prediction method by mapping the attack graph into an absorbing Markov chain, which not only ranked the threat levels of nodes but also quantified the probability distribution of attack paths with different lengths, but their method was not scientific for state transition probability calculation. Gong et al. [5] created a threat perspective by simply concatenating the detected assaults without considering the pre-post connection between devices and single-step attacks, which could only forecast the attack paths in simple circumstances. Yuan et al. [11] employed the breadth-first traversal algorithm in the attack path creation approach. The algorithmic model created all tracks in the attack scenario, resulting in path redundancy. A loop elimination algorithm was developed by Zhang et al. [12], which effectively avoided path redundancy and increased the effectiveness of threat path generation. However, they did not create inference rules because their ontology was only based on a graph database's search function, which could not explore the implicit knowledge. At the micro-layer: Wang et al. [13] evaluated the attack success likelihood. However, the attacker capability level was established without objective calculation findings

as a foundation, which might influence the prediction outcomes. Wu et al. [6], Zhang et al. [14] and Sun et al. [15] proposed the models can all predict and analyze attack paths from both macro and micro. Wu et al. [6] and Zhang et al. [14] did not consider factors affecting threat propagation direction when predicting paths, while Sun et al. [15] could not timely give defensive measures for the predicted threats.

In response to the above shortcomings of previous work, this paper proposes the two-layer model TL-TPM to predict the development trend of threat events at both macro and micro-layers. The macro-layer indicates the threat propagation path based on the knowledge graph. It examines both the attack success probability and the threat degree of each device, as well as combining the pre-and post-permissions to assess if the device is likely to be compromised. The micro-layer depicts the evolution process of the attack techniques based on the prediction results of the macro-layer and the temporal characteristics of the attack behavior, making the analysis more consistent with the actual situation of the network attack. The following are this paper's significant contributions:

1. Having studied the multi-source network security knowledge bases and integrated the information elements in them, the multi-source threat element analysis ontology and the network security knowledge inference method have been proposed to realize the association among heterogeneous network security knowledge bases.
2. Using the absorbing Markov chain as a bridge, we have innovatively mapped the attack graph to the event graph. At the same time, the Markov transition matrix is used to optimize the calculation of the event transition probability, making the attack process described by the attack graph can be more visually and accurately presented.
3. Proposing a two-layer attack prediction model, which combines the knowledge graph and event graph. It provides a comprehensive analysis of the evolution path of an attack from both macro and micro perspectives, visualizing the external trace and internal logic of the threat event development, which provides information and decision support for active defense.

2 Threat Modeling

2.1 Multi-Source Network Security Knowledge Integration and Ontology Construction

Different network security knowledge bases contain different kinds of information about threat events. To better integrate fragmented information for utilization, firstly, we collect, categorize, and organize information about threat events from network security knowledge bases. Secondly, we de-duplicate and fill in the gaps of the information to ensure the accuracy and completeness of them. Finally, the integrated information is classified and graduated to construct a complete ontology that enables fast and accurate access to relevant information for automated or semi-automated incident handling. The following are the knowledge bases utilized to collect information in this paper and Table 1 shows their specifics:

- Common Platform Enumeration (CPE) [16]
- Common Vulnerabilities and Exposures (CVE) [17]
- National Vulnerability Database (NVD) [18]
- Common Weakness Enumeration (CWE) [19]
- Common Attack Pattern Enumeration and Classification (CAPEC) [20]
- Adversarial Tactics, Techniques, and Common Knowledge Matrix (ATT&CK) [21]
- Detection, Denial, and Disruption Framework Empowering Network Defense (D3FEND) [22]

Table 1: Details of the knowledge bases

Knowledge base	Information of entries	Format of data	Linkage	First release date	Operator
CPE	Platform	XML	NVD	2007	NIST
CVE	Vulnerability	JSON	NVD	1999	MITRE
NVD	Vulnerability	JSON, XML	CVE, CWE, CPE	2005	NIST
CWE	Weakness	XML	CVE, NVD, CAPEC	2006	MITRE
CAPEC	Attack pattern	XML, CSV	CWE, ATT&CK	2007	MITRE
ATT&CK	Attack technique	JSON	CAPEC	2013	MITRE
D3FEND	Defense technique	CSV	ATT&CK	2021	MITRE

Fig. 1 depicts the relationships between the knowledge bases mentioned above. From these knowledge bases, we extract multi-source network security information and store it in a graph database. In particular, the items in each knowledge base function as nodes in the graph database, while the relational linkages across knowledge bases operate as edges. These edges are not bidirectional between the knowledge bases mentioned above. However, they can be bi-directionally navigated when incorporated into the graph structure. As a result, any node can be used to query the data in any knowledge base.

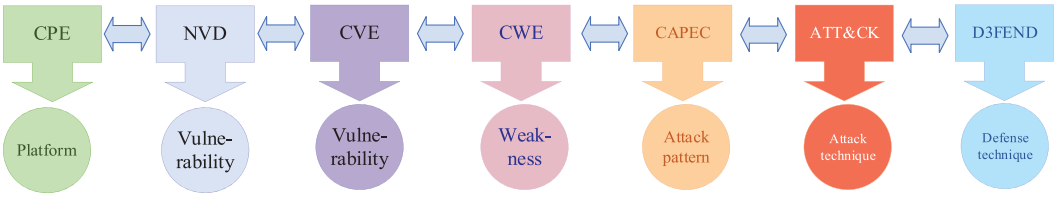

Figure 1: Linkages between knowledge bases

2.2 Classes and Attributes of MTEAO

We successfully linked multiple source knowledge bases and integrated the data from them as a source of security knowledge for developing our ontology model, the multi-source threat element analysis ontology (MTEAO). And we collectively call the entities in it, such as vulnerabilities, weaknesses, attack patterns, attack techniques, defense techniques, etc., as threat elements. The specifics of the MTEAO's classes are shown in Table 2.

The structure among classes is shown in Fig. 2a, while the logical links among the second-level subclasses are shown in Fig. 2b.

Table 2: The details of the classes

Top-level class	Second-level subclass	Attribute	Source of knowledge
Defend	Artifact, Defend_Technique	hasArtifact, beAgainst, hasMitigation	D3FEND
Attack	ATTCK_Technique, AttackPattern, Weakness, Vulnerability	hasArtifact, beAffected, beExploited, beUsed, belong_to, hasCVSS, hasLevel, useATechnology	ATT&CK, CAPEC, CWE, CVE, NVD
Attacker	Attacker_Name	useATechnology, beAgainst, hasAccess, hasCompromised	ATT&CK
Device	Device_Name, Asset	beAffected, hasAsset, hasAccess, hasPreRoute, hasCompromised	CPE

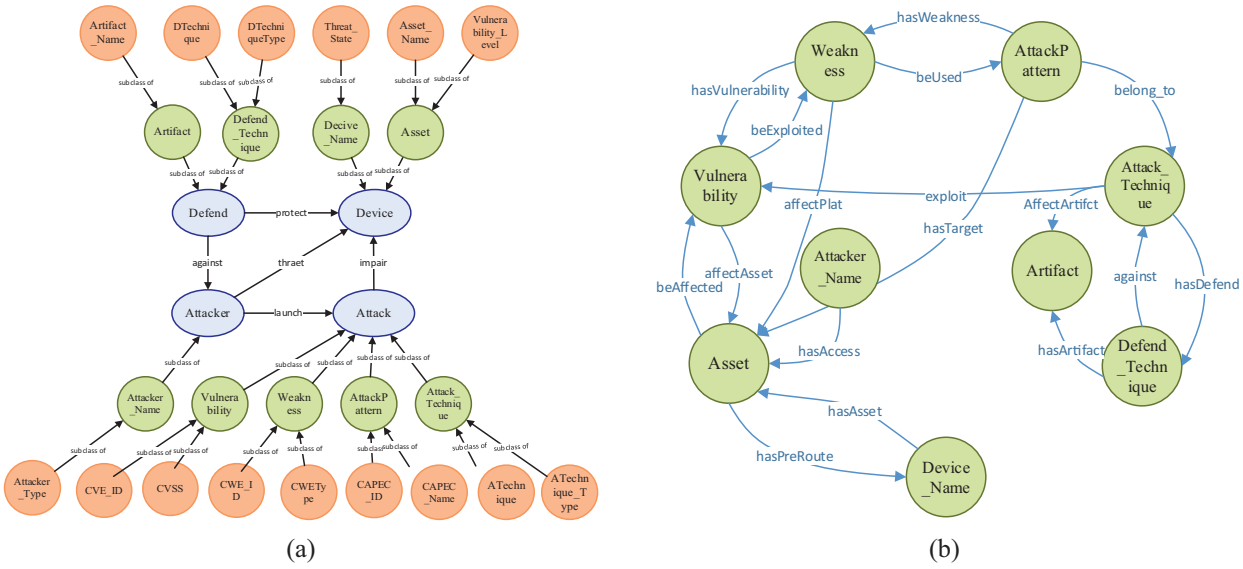

Figure 2: (a) The inclusion relationships among classes; (b) The logical links among the second-level subclasses

2.3 Inference Rules of MTEAO

2.3.1 Design of Inference Rules

Using inference rules enables us to deduce possible knowledge based on existing information, allowing us to discover new implicit correlations between threat elements. Protégé's inference engine can execute sequential multi-step inference and aids in comprehending the inferred findings via inference interpretation. Table 3 shows how the seven inference rules in this paper are intended to serve diverse purposes.

Table 3: Inference rules and usages

R_1	ATechnique(?t)^hasArtifact(?t, ?a)^Artifact(?a)^hasMitigation(?a, ?d)→hasDefend(?t, ?d)
Usage	The attack technique can be mapped to the defense technique by digital artifact. If the security staff knows the attack technique employed by the attacker, they may immediately receive the defense approach that matches it.
R_2	Artifact(?f) ^ DTechnique(?d) ^ hasMitigation(?f, ?d) ^ sameAs(?f, EventLog)→sqwrl:select(?f, ?d)
Usage	Each digital artifact is connected with its related defense technique, and various digital artifacts may be at varying degrees of risk. Security staff can prioritize query defense techniques for the most susceptible digital artifact.
R_3	Asset(?p) ^ beAffected(?p, ?v) ^ CVE_ID(?v) ^ hasCVSS(?v, ?s) ^ CVSS(?s) ^ hasLevel(?s, ?l) ^ VulnerableLevel(?l) →hasSeverityLevel(?p, ?l)
Usage	The identified asset has a vulnerability, and severity levels categorize different types of vulnerabilities. The vulnerability level of the asset is determined by the severity of the identified vulnerability. CVSS scores quantitatively assess the severity of vulnerabilities, and they are divided into five levels: “Critical”, “High”, “Low”, “Medium”, and “None”. According to the inference rule, “Critical” and “High” are automatically assigned to the asset’s high vulnerability level “HighLevel,” “Medium” is assigned to the asset’s medium vulnerability level “MediumLevel,” and “Low” and “None” are assigned to the asset’s low vulnerability level “LowLevel.”
R_4	Asset(?p) ^ VulnerableLevel(?l) ^ hasSeverityLevel(?p, ?l) ^ sameAs(?l,HighLevel)→ sqwrl:select(?p,?l)
Usage	If the asset has a high vulnerability level, running the query will identify it. Security staff can easily locate it and give it priority for maintenance.
R_5	Asset(?t) ^ beAffected(?t, ?y) ^ Vulnerability(?y) ^ beExploited(?y,?s) ^ Weakness(?s) ^ sameAs(?s, CWE-22)^ AttackPattern(?n) ^beUsed(?s, ?n) ^ belong_to(?n, ?u) ^ ATechnique(?u) ^ sameAs(?u, T1027) → sqwrl:select (?t, ?u)
Usage	The vulnerability in assets is often linked to a particular weakness, which in turn is associated with an attack pattern, and that pattern is linked to a specific attack technique. By running a query, one can identify the asset that has a particular weakness and is affected by the specific attack technique. This enables security personnel to isolate device that has specific weakness and is under attack by specific attack technique.
R_6	Device(?e) ^hasAsset(?e,?t) ^Asset(?t) ^beAffected(?t,?y) ^Vulnerability(?y) ^beExploited(?y, ?w) ^ Weakness(?s) ^ beUsed(?s, ?n) ^ AttackPattern(?n) ^ belong_to(?n, ?u) ^ ATechnique(?u) ^ hasArtifact(?u, ?c) ^ Artifact(?c) ^ hasMitigation(?c, ?i)→ sqwrl:select(?e, ?t, ?y, ?s, ?n, ?u, ?c, ?i)

(Continued)

Table 3 (continued)

Usage	The device can sequentially identify the asset, vulnerability, weakness, attack pattern, attack technique, digital artifact, and defense technique connected to it. When the query is initiated, all relevant threat elements are presented. Security personnel have the option to look through all the results or focus on specific ones to address potential threats.
R_7	Attacker(?r) ^ Device(?e) ^ Asset(?t) ^ hasAsset(?e, ?t) ^ Vulnerability(?y) ^ beAffected(?t, ?y) ^ sameAs(?yy, CVE-2021-31645) ^ hasPreRoute(?t, FTPS) → hasCompromised(?r, ?t) ^ hasAccess(?r, ?e)
Usage	If the device has an asset with a vulnerability and the previous neighboring device is connected to it with a device access path, the attacker can compromise the device and harm the asset. Additionally, by using inference rules R6 and R7 together, it becomes possible to track the vulnerability, weakness, attack pattern, attack technique, and defense measure of the compromised device over time.

2.3.2 Application of Inference Rules

Then, we will demonstrate the practical application of inference rules in combating security threats. Below are two distinct scenarios that will showcase their effectiveness:

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

The asset "arch_newsworld" is stored in the email server and has a vulnerability known as "CVE-2005-3435" with a severity level of "High". In Fig. 3, the green box shows the vulnerability level of "arch_newsworld" is "HighLevel" by executing inference rule "R_3". Additionally, the officer can use the inference rule "R_7" to determine if an attacker can conquer the asset. The red box shows that the attacker can obtain complete control of the email server and compromise the "arch_newsworld" asset.

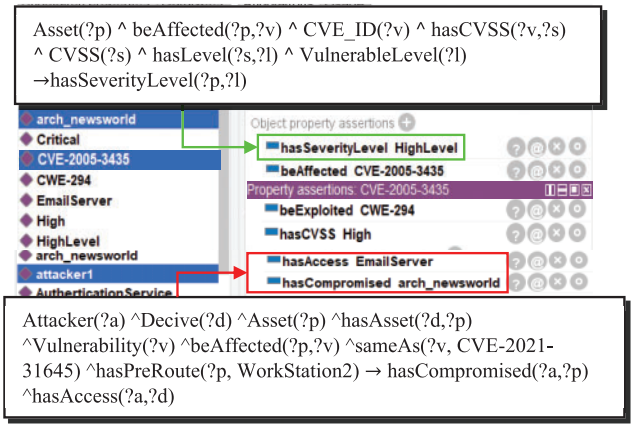

Figure 3: The result of determining the vulnerability level of the asset and whether the asset will be conquered

2. Search for information on attack and defense

The security officer can execute the "R_6" inference rule to retrieve information on devices, assets, vulnerabilities, weaknesses, attack patterns, attack techniques, digital artifacts, and defense techniques. Displayed in the yellow box is the output of utilizing "R_6", as depicted in Fig. 4. The "R_1" inference rule can be used by the security officer to search defense techniques that relate to specific attack techniques directly. The green box displays the defense techniques for the attack technique "T1211".

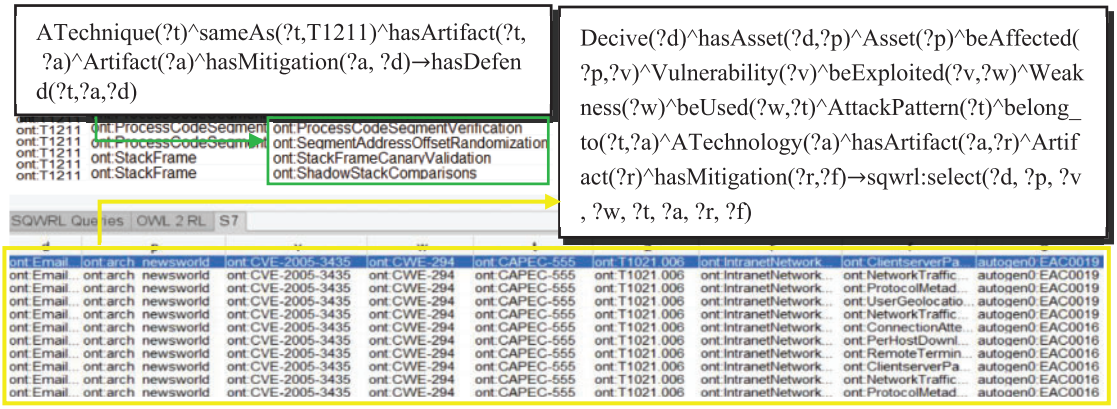

Figure 4: The result of searching for information on attack and defense

3 Two-Layer Threat Prediction Model

In the event of a system threat, the top priority is to address and contain it promptly. As a result, it is crucial to evaluate and forecast the potential progression of the threat. This paper proposes a two-layer threat prediction model called TL-TPM, which aims to enhance the accuracy of predicting attacks. The macro-layer of TL-TPM draws the propagation path of threat between devices and associates these devices with the corresponding threat elements for threat alerting and response; The micro-layer depicts the evolution process of attack techniques while warning of attack techniques with a high probability of use, assisting security personnel to strengthen the prevention of specific attacks. The workflow of this paper is shown in Fig. 5.

3.1 The Macro-Layer of Threat Prediction Based on Knowledge Graph

To accomplish his attack goal, the attacker will exploit weaknesses in the target network and execute a series of consecutive attacks. The macro-layer of TL-TPM maps this set of attack sequences as a propagation path of threat between devices. We describe the concept "attack" in the ontology as a relation link between the attackers and devices, changing the attack path prediction issue into the link prediction issue in the knowledge graph. To aid in the explanation of the below algorithm, the appropriate definitions are provided:

- Core asset (cas): The target asset the attacker aims to seize or obliterate.
- Threat degree (thd): The level of risk to the core asset when the device is under attack. The greater the threat level of a device, the more likely it is that an attacker will select that device for the next attack, leading the threat to spread to the core asset. thd $\in$ [0, 1].
- Threat degree interval (tdi): Security personnel determine the interval of threat degree to classify the risk stages according to their needs.
- Topology layer (tl): The positioning layer of a device in the system topology. The device closer to the core assets is defined as a higher layer.

- Attack success probability (asp): The success probability of an attacker performing a single-step attack.
- Device set (Devices): A set of all the devices in the system.
- Business access relationship (bar): The access and control relationship between two devices. $d_0 \overset{bar_1}{\rightarrow} d_1 \overset{bar_2}{\rightarrow} \cdots \overset{bar_n}{\rightarrow} d_n$ expresses the business access relationships from device d_0 to device d_n. And Bar denotes the set of business access relationships.
- Device access path(dpath): An acyclic series of devices connected by business access relationships. The device access path from the specific device d_0 to the core asset located device d_n is represented as $dpath = \{d_0, d_1, ..., d_n\}$.
- Threat propagation path (tpath): It is an ordered sequence of devices conquered by the attacker.
- Initial device (ind): The device initially attacked by the attacker.
- Pre-privilege: It is the pre-condition that a business access relationship exists between device d_t and the previous one d_{t-1}.
- Post-privilege: It is the post-condition that there is a vulnerability in the device, leading the attacker to gain complete control of the device d_t by launching an attack.

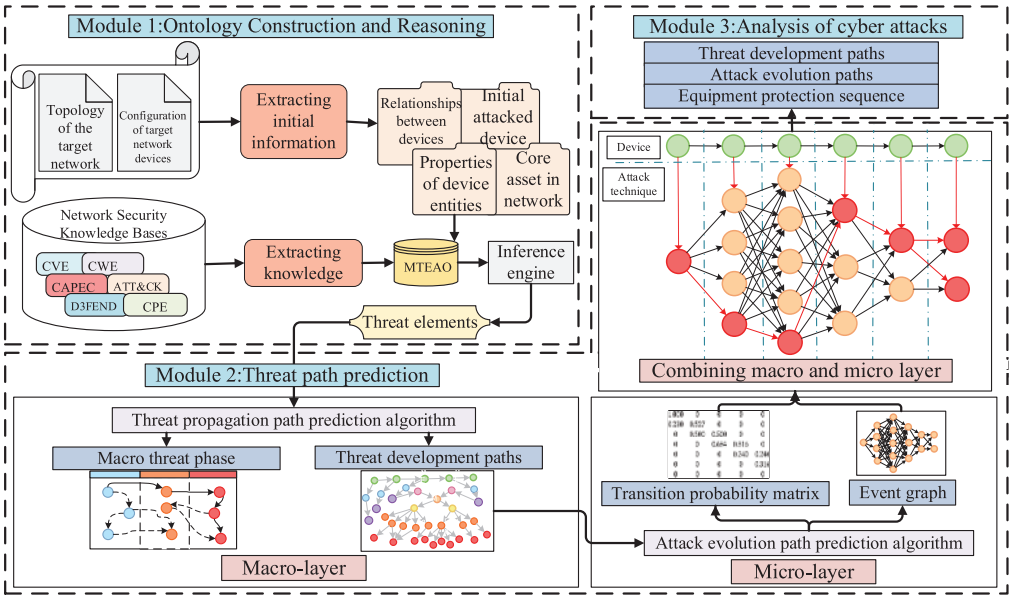

Figure 5: Workflow of the system

3.1.1 Calculation of Threat Influence Elements

The role of the attacker's psychology in the threat spread procedure is overlooked by most existing attack prediction systems. We evaluate the threat degree of the device based on the attack success probability to estimate the threat propagation path, considering that an attacker would always use the most favorable methods to attack the most susceptible device.

1. Calculation of the Attack Success Probability

Attack success probability refers to the success probability of an attacker performing a single-step attack. Specifically, there are two types of attacks: social engineering attacks and vulnerability exploit attacks. Professional security staff can easily avoid social engineering attacks, so the probability of

success is low at 0.2. While the probability of success for vulnerability exploit attacks is determined by the Common Vulnerability Scoring System (CVSS) score [23].

The CVSS score has a base score (Base) that reflects the inherent characteristic of a vulnerability, which remains unchanged over time and environment. The composition of the CVSS score is shown in Table 4. And its calculation formulae are shown in Eqs. (1) and (2).

Table 4: Composition of the CVSS score

Score	Sub-score	Second-sub score	Purpose
Base	Exploitability subscore (ESC)	Attack vector (AV) Attack complexity (AC) Privilege required (PR) User interaction (UI)	Measuring the ease of vulnerability exploitation
	Impact subscore (ISC)	Confidentiality impact ($Impact_{Conf}$, I_C) Integrity impact ($Impact_{Integ}$, I_I) Availability impact ($Impact_{Avail}$, I_A)	Measuring the harm of a vulnerability

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

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

when the vulnerability code is more mature, there is a greater chance that the vulnerability will be successfully exploited. So, we add the code maturity (ExploitCodeMaturity) to optimize the score [15], which is multiplied by 0.1 to represent the attack success probability. The attack success probability is calculated as Eq. (3).

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

2. Calculation of the Threat Degree

If device d_0 in the device access path *dpath* is compromised, the threat degree to the core asset can be calculated in the following way:

i. When $dpath = \{d_0\}$, which means that the core asset is present in the initial device of the path, and that device has been compromised, then the threat degree is computed as Eq. (4).

$$dht_{(d_0,\ cas)} = 1 \tag{4}$$

ii. When $dpath \neq \{d_0\}$, the attacker can only spread the threat from one device to another by conducting an attack. Therefore, the threat degree of the device can only be determined if the threat propagation path *tpath* exists on the device access path *dpath*. If not, it signifies that

the threat cannot be disseminated to the core asset via the *dpath* by attack techniques. As a consequence, the threat degree is 0.

To successfully compromise the high-topology layer device, the low-topology layer device must first be compromised. Consider the ratio of topological layer numbers between the device and the core asset as the weight. A higher weight indicates that device d_t is closer to the core asset. In addition, the attacker must take control of every device in the threat propagation path before device d_t if he wishes to compromise 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 initial device d_0 to device d_t. In this case, the threat degree is calculated as Eq. (5).

$$thd_{(d_t,cas)} = \begin{cases} \frac{tl_{dt}}{tl_{cas}} * \left(\prod_{d \epsilon tpath} asp\,(d)\right), tpath \neq \varnothing \\ 0, \qquad\qquad\qquad tpath = \varnothing \end{cases} \tag{5}$$

If device d_t has more than one adjacent device, and there is the *tpath* on the *dpath* between d_t and each adjacent device. Then, the device with the highest threat degree among the adjacent devices is selected as the next target to attack and spread the threat.

3.1.2 Threat Propagation Path Prediction Algorithm

Next, this paper presents the threat propagation path prediction algorithm (TPPPA) based on the knowledge graph. TPPPA not only sequentially strings the attacked device nodes into a path but also associates them with the corresponding multi-source threats elements. It predicts the path while outputting relevant threat information, giving security personnel an intuitive understanding of the attacks being suffered and their countermeasures.

The core code of TPPPA is as follows:

Input: *Devices*, *Bar*, *TDI*, *cas*, *ind*, *MTEAO*

1) Initialize *DPaths*, *tpath*, *Privileges*, *AS*;
2) *target*=extractAss(*cas*,*Decives*); // Search for the device *target* where the core asset *cas* is held in the device set *Devices*.
3) *DPaths*=Pathgenerate(*ind*,*target*,*Devices*,*Bar*); // String devices into device access paths via business access relation-ships.
4) *prePrivileges*,postPrivileges = attReason(*ind*,*Bar*,*MTEAO*); // Extract the pre-and post-privileges from the inference results and respectively place *them in the prePrivileges* and *postPrivileges*.
5) function Iteration(*ind*): // The function of the iterative attack.
6) tpath.append(*ind*);

(Continued)

(continued)

```
7)  Initialize degrees;                                   // Initialize the set of threat degrees for all adjacent devices at the next layer.
8)   localDevices = findLocal(ind,prePrivileges)          // Find all adjacent devices at the next layer based on the prePrivileges.
9)   for local in localDevices:                           // Find the next target device local in the adjacent device set localDevices.
10)   if extractPost(local, postPrivileges) == true:      // If the device's post-privilege can be found in the post-Privileges.
11)    degree.append(calculateThd(local, tpath, target))  // Calculate the threat degree of all adjacent devices.
12)   end if;
13)   else:
14)    degree.append(0);
15)   end else;
16)  end for;
17)  maxd = maxDevice(degree, localDevices)               // Extract the device with the highest threat degree among all adjacent devices to be the next target device.
18)  if maxd != target:
19)   Iteration(maxd);
20)  end if;
21) end;
22) GeneratClusterGraph(Dpaths,tpath,TDI)                 // Cluster devices into corresponding threat degree intervals to construct a macro risk state graph (MRS).
23) Iteration(ind);
24) for device in tpath:
25)   tpath.append(attScenario(device));                  // Query the threat elements for all devices in tpath.
26) end for;
```

Output: macro risk state graph *MRS*; threat propagation path *tpath*

The algorithm described above follows a series of steps: Steps 1)∼4) involve initializing the required sets and extracting required data. Steps 5)∼21) form the heart of the algorithm, predicting the threat propagation path. Because to completely control the device d_t, both requirements must be simultaneously met: 1. The adjacent device d_{t-1} of device d_t had been completely controlled by the attacker. And a device access path exists between the d_{t-1} and d_t. 2. The device d_t contains a vulnerability. As a result, the pre-privileges are extracted to establish the device access path and then the post-privileges are extracted to determine whether the threat propagation paths exist. Then, select

the device with the highest threat degree among the adjacent devices as the next target. Afterward, create the directed edge to form the complete threat propagation path. Step 22) clusters devices into corresponding threat degree intervals and produces the macro threat state graph. Steps 23) ∼26) extract threat elements for all devices in the *tpath* and presents the result with the knowledge graph.

3.2 The Micro-Layer of Threat Prediction Based on Event Graph

The micro-layer of TL-TPM uses an absorbing Markov chain to map the attack graph to the event graph, which depicts the evolution process of attack techniques. It can warn of attack techniques with a high probability of use, assisting security personnel to strengthen the prevention of specific attacks.

3.2.1 Preliminary Knowledge and Theoretical Arguments

This section first explains the basic concepts of the attack graph and absorbing Markov chain, then argues the rationality of mapping the attack graph to the event graph through the absorbing Markov chain. Finally, the attack evolution path prediction algorithm is presented.

1. Attack graph

The attack graph (AG) is a visualization method to model the association of multi-step attack behavior and represent the attack process [24]. It is a directed graph that portrays all possible penetration paths of an attacker in the network. An example of an attack graph is shown in Fig. 6.

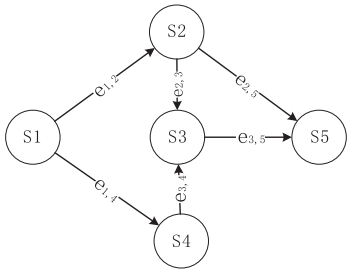

Figure 6: The example of an attack graph

AG is represented by a quadruple $AG = (S, E, A, \delta)$, where:

- S denotes the set of state nodes, $S = \{S_i | i = 1, 2, \cdots, j\}$ denotes the set composed of j different state nodes, and the state nodes can be divided into starting state nodes, transition state nodes, and target state nodes. For example, in Fig. 6, S_1 is the start state node, S_2, S_3, S_4 are the transition state nodes and S_5 is the target state node;
- E denotes the set of directed edges between state nodes, $e_{m,n} \in E$, $e_{m,n}$ represents the edge of the state node S_m pointing to S_n, i.e., a state transfer has occurred from S_m to S_n;
- A denotes the set of atomic attack nodes, $A = \{a_i | i = 1, 2, \cdots, j\}$, a_i is an atomic attack, with each successful attack corresponding to a state transition $e_{m,n}$.
- δ denotes the set of state transition probabilities, $\delta(e_{m,n})$ denotes the probability $P_{(S_m|S_n)}$ of the attacker transferring from state S_m to state S_n, and $\delta(e_{m,n})$ is equal to the probability $P_{(a)}$ of an atomic attack a occurring.

2. Absorbing Markov chain

The main advantages of Markov processes are the ability to build prediction models in time based on statistical information or the results of operational observations [25]. And the Markov chain (MC)

is a Markov process in which both time and state are discrete [26]. For a discrete set $S = \{s_1, s_2, \cdots, s_n\}$ containing a finite number of states, each state is only related to the previous adjacent state, called posteriority-free, i.e., $P_{(s_i|s_{i-1},s_{i-2},\cdots s_1)} = P_{(s_i|s_{i-1})}$. The probability $P_{(s_i|s_{i-1})}$ is the transition probability of the state $s_i \rightarrow s_{i-1}$, and the transition probabilities between all state nodes form the state transition probability matrix P.

The absorbing Markov chain (AMC) is an MC that contains at least one absorbing state and from which any one of the states can eventually reach the absorbing state. If an AMC has r absorbing states, t non-absorbing states, and all states are n, then $n = r+t$. At this point, the state transition probability matrix is expressed as $P = \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix}$. Q is the $t \times t$ matrix representing the probabilities of transition between transition states; 0 is the $r \times t$ zero matrix; R is the $t \times r$ non-zero matrix representing the transition probabilities from transition states to absorbing states; and I is the $r \times r$ unit matrix.

3. Mapping of attack graph to the absorbing Markov chain

In AG, the transition of the current state s_i to the next state s_{i+1} is only related to whether the state s_i satisfies the vulnerability exploitation, independent of the previous states, at which point the transition between states is precisely in line with the posteriority-free property of MC; The attacker will eventually reach a stable termination state through a multi-step attack based on vulnerability exploitation, which is consistent with the absorption state of AMC; A network attack has at least one termination state, and an AMC has at least one absorbing state; And the successful probability of atomic attack in AG can be regarded as the state transition probability in AMC. Therefore, AG can be mapped to AMC.

4. Mapping of absorbing Markov chain to event graph

The event graph (EG) represents events and their relationships as a logically directed graph. It takes abstract and generalized events as nodes, connected to form directed edges that express the evolution process between events. And this process can be considered as a transition between events, then the transition probability on the directed edge represents the probability of the event's evolution. This probability can be calculated and expressed precisely in terms of the transition matrix of AMC. Thus, AMC can be mapped to the EG. At the same time, we can optimize the Markov transition matrix by considering multiple dimensions affecting the event transition and assigning different weights to them. So far, we have achieved the mapping from AG to EG.

3.2.2 Attack Evolution Path Prediction Algorithm

Unlike the way of calculating event transition probability in general EG, this paper optimizes it to reflect the event evolution process better. We propose an available method for measuring the hazard of an attack technique. We calculate the hazard of attack techniques from three metrics: "Life Cycle Stage", "Likelihood of Attack", and "Skills Required0". The higher hazard means the higher the probability that the attacker will use the attack technique, then the higher the likelihood that the attack technique will transfer.

The ATT&CK matrix contains 14 attack strategies, and each attack strategy includes several attack techniques. It represents a complete sequence of attack lifecycle stages in the form of a table from left to right. The further back the attack technique is in the lifecycle stage, the closer it is to complete an attack and the more harmful it is. Therefore, each attack technique is scored according to the attack lifecycle stage it belongs to.

The two metrics in CAPEC are: "Likelihood of Attack" and "Skills Required". Both metrics measure the probability of an attack occurring and are graded as "High", "Medium", and "Low".

As shown in Table 5, we converted them into scores "9," "6", and "3" to quantify the probability of using the attack technique. The higher the probability that an attack technique is used, the more harmful it is.

Table 5: Grade and score

Metrics	Grade	Score
1) Likelihood of Attack	High	9
2) Skills Required	Medium	6
	Low	3

Each attack technique is scored on the above three metrics, and the three scores are summed and averaged for the final attack technique hazard score. Based on the method of attack technique hazard metric, we propose the attack evolution path prediction algorithm (AEPPA). AEPPA normalizes the attack technique hazard score to realize the mapping from AG to AMC and finally constructs the EG with the Markov transition matrix. The core code of AEPPA is as follows:

Input: the set of attack techniques hazard ATH, threat propagation path *tpath*

```
1)  Initialize att_amc;                          // Initialize the list for transition probability matric of AMC.
2)  dt_s = getScore(tpath, DT);                  // Get hazard scores for all attack techniques.
3)  att_m = createAttM(dt_s)                     // Generate an n×n dimensional attack technique hazard scores matrix.
4)  for row in att_m.get_rows():                 // Iterate over each row in the matrix.
5)      r_total= row.totals()                    // Sum the values of the elements in row i.
6)      r_new = [ ]                              // Generate a new row.
7)      for num in row:                          // Iterate over each element in row i.
8)          r_new.append(num / r_total)          // Calculate the element's value in row i, column j, and put it in the new position of row i, column j.
9)      att_amc.append(r_new)                    // Generate transition probability matrix.
10) generateEventicGraph(att_amc);               // Obtain the EG based on the transition probability matrix.
```

Output: hazard score for each attack technique; transition probability matrix; attack technique evolution event graph

The algorithm described above follows a series of steps: Step (1) initializes the list for transition probability matric of AMC. Step (2) uses the method of attack technique hazard metric to obtain the hazard scores for all attack techniques based on the set of attack techniques hazards. Step (3) generates an n × n dimensional matrix using the hazard scores of all attack techniques. If the attacker and the attack techniques are considered state nodes, then n represents the number of state nodes, and the values of row_i represents the score from the $statenode_i$ to all state nodes. Steps (4) ~ Step (9) calculate and obtain the transition probability of each attack technique to itself and any other attack technique, and put them into the matrix list in Step (1). At this point, we obtain the transition probability matrix of AMC. Step (10) gets the EG based on the transition probability matrix.

AEPPA finally outputs the hazard score for each attack technique and the transition probability matrix of the attack techniques, enabling the subsequent analysis of the evolution process to depend on accurate data. At the same time, the visualization of EG enhances the understanding of the evolution process of threat events.

4 Experiment

4.1 Scene of the Experiment

The experiment scene is shown in Fig. 7. The system consists of three subnets, with a firewall and the intrusion detection systems (IDS) deployed to achieve access control and intrusion detection. The firewall allows only the workstation and web server in the demilitarized zone (DMZ) to interact with the outside world, and the network line of the workstation1 is connected from the router; Subnet 1 deploys an administration station, a web server, and a file transfer protocol server. And the router also connects with the administration station, which can interact with workstation1 and access the web server2 and file transfer protocol server; Subnet 2 deploys a workstation and a data server. Web server1 and workstation2 have user accounts of the data server and can access the data server. Tables 6 and 7 present the corresponding information and the business access relationships of the devices in the system.

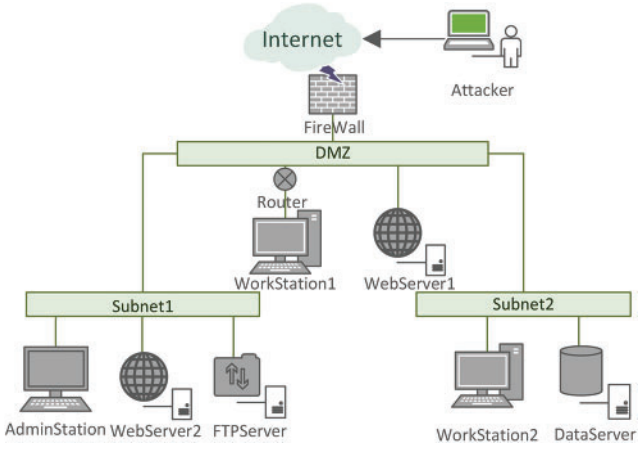

Figure 7: Scene of the experiment

Table 6: Instance details

Device	Topology hierarchy	Asset	Vulnerability	CVSS	Vulnerability type
Firewall	1	wordfence_ security	CVE-2022-3144	4.8	Cross-site scripting
Router	2	rv_110 w	CVE-2022-20923	9.8	Bypass a restriction or similar
Web server1	2	phpBB	CVE-2005-0603	5.0	Obtain information

(Continued)

Table 6 (continued)

Device	Topology hierarchy	Asset	Vulnerability	CVSS	Vulnerability type
Workstation1	3	vcenter_server	CVE-2021-21972	9.8	Execute code directory traversal
Data server	3	SQL	CVE-2004-0366	7.5	SQL injection
Workstation2	3	matrix_screen_saver	CVE-1999-1454	4.6	Bypass a restriction or similar
Admin station	4	WeCube	CVE-2022-37785	7.5	Privilege escalation
Web server2	5	mac_os_x	CVE-	5.8	Privilege escalation
FTP server	5	glFTPd	CVE-2021-31645	7.5	Denial of service

Table 7: Business access relationships

From	To	Object relationship
Firewall	Router (rv_110 w)	hasRoute (FW, rv_110 w)
Firewall	Web server1 (phpBB)	hasRoute (FW, phpBB)
Router	Workstation1 (vcenter_server)	hasRoute (Router, vcenter_server)
Router	Admin station (google_chrome)	hasRoute (Router, google_chrome)
Web server1	WS_2 (matrix_screen_saver)	hasRoute (Web_1, matrix_screen_saver)
Workstation1	Admin station (google_chrome)	hasRoute (WS_1, google_chrome)
Workstation2	Data server (SQL)	hasRoute (WS_2, SQL)
Admin station	Web server2 (mac_os_x)	hasRoute (AS, mac_os_x)
Admin station	FTP server (glFTPd)	hasRoute (AS, glFTPd)
Web server2	FTP server (glFTPd)	hasRoute (Web_2, glFTPd)
Data server	FTP server (glFTPd)	hasRoute (DS, glFTPd)
FTP server	—	—

4.2 Threat Prediction

The following initial conditions are given in Table 8 according to the experiment scene. In this section, predictions are respectively made at the macro-layer and micro-layer.

Table 8: Initial conditions

From	Attack device	Device vulnerability	Has core asset
Attacker	Firewall	CVE-2022-3144	FTP Server

4.2.1 Macro Threat Prediction Experiment Based on Knowledge Graph

Based on the threat degrees of the devices, users can set the appropriate threat degree intervals according to their needs to divide the threat status stages and cluster devices with the same threat degree in the same interval. Assume that the enterprise stipulates that the threat degree does not exceed 0.15 is low-risk status, 0.15 to 0.20 is medium-risk status, and over 0.20 is high-risk status. And the three risk states of low, medium, and high are respectively marked with blue, yellow, and red colors. Executing the TPPPA based on the initial conditions, the macro risk state graph is constructed as shown in Fig. 8. The circles in Fig. 8 represent the devices under attack; the dotted links constitute the device access paths; and the solid links form the threat propagation path, indicating the actual trajectory of the threat as it moves from the low threat degree devices to the high threat degree devices. The threat degree and vulnerability type of each device is shown in the rectangular box. For simplicity of expression, the devices are replaced by abbreviations, e.g., the firewall is written as FW.

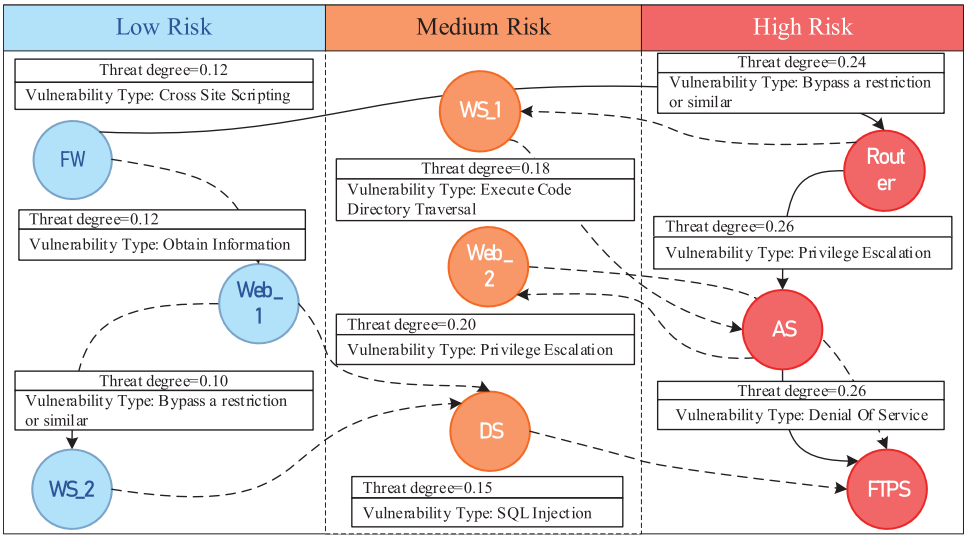

Figure 8: Macro risk state graph

At the same time, TPPPA calculates the devices most likely to be compromised by the attacker at each step, links them sequentially into the path, and connects them to the associated multi-source threat elements for a complete threat propagation path graph. The threat propagation path is marked with black arrows in Fig. 9, and the different colored circles represent different threat elements. Security personnel can rely on the graph to quickly grasp the threat and take appropriate defensive measures for each attack to contain the spread of the threat.

Based on the experiment results, the attacker's intent was analyzed as follows:

1. The attacker conquered Firewall (FW) by attacking the vulnerability "CVE-2022-3144" in the software "Wordfence_Security", which caused FW to be injected malicious web scripts into the settings and to be compromised completely.
2. Then, the attacker attacked the Router by exploiting the vulnerability "CVE-2022-20923" in hardware "rv_110w", which allowed the unauthenticated attacker to bypass authentication.
3. Since Work Station 1 (WS_1) was connected from the Router and its server management software "vcenter_server" contained a remote code execution vulnerability "CVE-2021-21972".

The attacker used CVE-2021-21972 to execute commands with unrestricted privileges and thus gained complete control of WS_1.

4. There was a business access path between WS_1 and the Admin Station (AS), and the attacker attacked the AS along the network. AS owned the software "WeCube", which contained the vulnerability "CVE-2022-37785" that caused plaintext passwords to be displayed in the terminal plug-in configuration. The attacker then exploited the vulnerability to steal passwords and gain complete control of AS.
5. Via AS, the attacker accessed the FTP Server (FTPS), where the core asset is located. The FTPS contained the software "glFTPd" with the vulnerability "CVE-2021-31645". By breaking the link limit with CVE-2021-31645, the attacker triggered a threat of denial service.

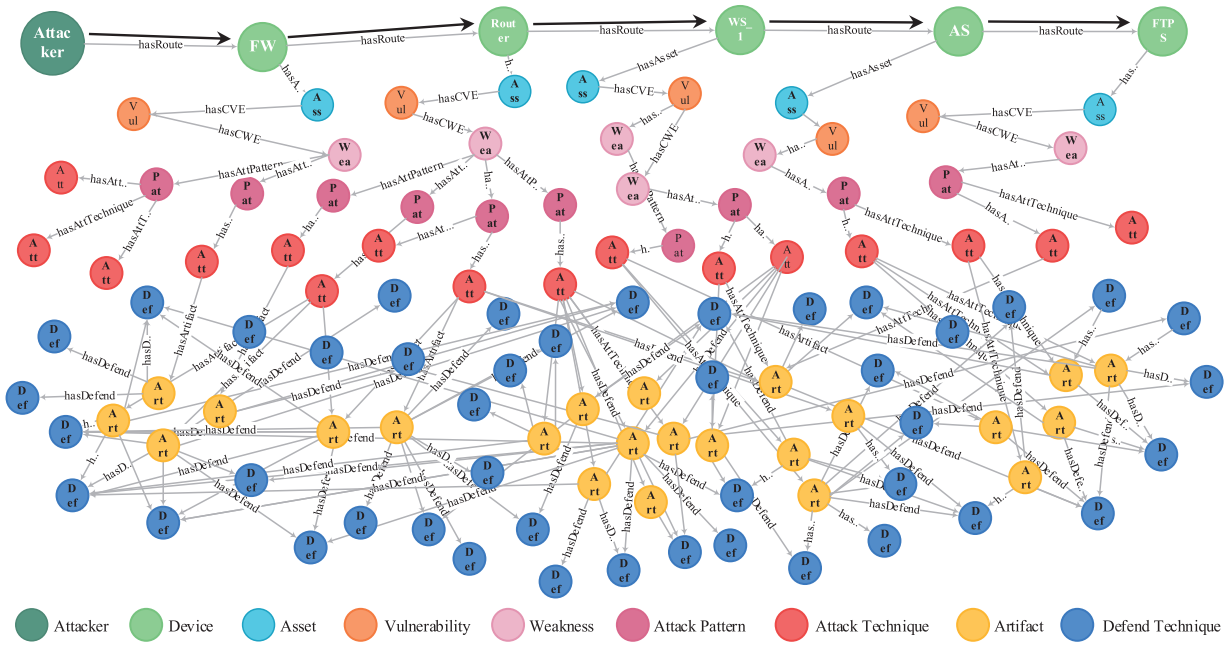

Figure 9: Threat propagation path and threat elements

Combined with the macro risk state graph, the experiment results were compiled to present the corresponding prediction information, as shown in Table 9.

Table 9: Prediction of the threat propagation path

Device	Risk state	Threat degree	Be the next target device	The next target device	Current threat propagation path
FW	Low	0.12	Yes	Router	Attacker→FW
Router	High	0.24	Yes	WS_1	Attacker→FW→Router
Web_1	Low	0.12	No	—	—
WS_1	Medium	0.18	Yes	AS	Attacker→FW→Router→WS_1

(Continued)

Table 9 (continued)

Device	Risk state	Threat degree	Be the next target device	The next target device	Current threat propagation path
WS_2	Low	0.10	No	—	—
DS	Medium	0.15	No	—	—
AS	High	0.26	Yes	FTPS	Attacker→FW→Router→WS_1→AS
Web_2	Medium	0.20	No	—	—
FTPS	High	0.26	—	—	Attacker→FW→Router→WS_1→AS→FTPS

Through the above analysis, the attack steps can be visualized, and the predicted threat propagation path can be used to contain the threat spread in time, which proves the effectiveness and practicality of TPPPA. While TPPPA is based on the ontology model MTEAO, this ontology model extends and improves the modeling knowledge of the security domain compared to the previous work. In Table 10, the MTEAO is compared to other ontology models, and the results are presented below:

Table 10: Comparison among the network security ontology models

Ontology	Asset	Vulnerability	Weakness	Attack pattern	Attack technique	Defend technique	Support inference
Wu et al. [6]	√	√	×	×	√	×	√
Iannacone et al. [7]	×	√	×	×	√	×	×
Syed et al. [8]	√	√	×	√	√	×	√
Yuan et al. [11]	×	√	×	×	×	×	×
Zhang et al. [12]	×	√	×	×	×	×	×
Sun et al. [15]	√	√	×	×	√	×	√
MTEAO	√	√	√	√	√	√	√

4.2.2 Micro Threat Prediction Experiment Based on Event Graph

Based on the prediction results of the threat propagation path in Experiment 4.2.1 and executing AEPPA, the attack technique hazard scores of the devices, the Markov transition probability matrix, and the attack technique evolution event graph are obtained to deepen the prediction.

AEPPA first takes the path predicted by TPPPA as input and outputs the state transition matrices P and Q, then calculates the matrix N according to the formula $N = (I-Q)^{-1}$. The matrix N represents the expected number distribution of state node visits. The values in the first row of it are the number of visits from the state node S_1 to each remaining state node. In the context of the experiment in this paper, the values in the first row of the matrix N can be interpreted as the number of times the attacker uses each attack technique. And the higher the number of times the attack technique is used, the higher

the probability of its use. Matrices P, Q, and N are shown below. Tables 11 and 12 give information about the attack techniques based on the results returned by TEPPA.

Table 11: The hazard scores of attack techniques

Device	State	Attack technique	Life cycle stage	Likelihood of attack	Skills required	Hazard score
FW	S_2	T1082	Discovery	Low	Low	5.00
	S_3	T1592	Reconnaissance	High	Low	4.33
	S_4	T1595	Reconnaissance	High	Low	4.33
	S_5	T1217	Discovery	High	Low	7.00
Router	S_6	T1040	Credential access, discovery	Medium	Medium	5.83
	S_7	T1548	Privilege escalation, Defense evasion	Medium	Medium	6.17
	S_8	T1550	Defense evasion, Lateral movement	High	Low	6.83
	S_9	T1185	Collection	High	Low	7.67
	S_{10}	T1557	Credential access, Collection	High	Medium	8.17
WS_1	S_{11}	T1027	Defense evasion	High	Medium	7.33
	S_{12}	T1134	Defense evasion, Privilege escalation	High	Low	6.17
	S_{13}	T1528	Credential access	High	Low	6.67
AS_2	S_{14}	T1005	Collection	High	High	8.67
	S_{15}	T1552	Credential access	High	Medium	7.67
FTPS	S_{16}	T1498	Impact	Low	Low	6.67
	S_{17}	T1499	Impact	Low	Low	6.67

Table 12: The transition probabilities of attack techniques

State transition	Attack transition	Transition probability	State transition	Attack transition	Transition probability	State transition	Attack transition	Transition probability
$S_1 \rightarrow S_2$	Attacker→T1082	0.242	$S_4 \rightarrow S_9$	T1595→T1185	0.245	$S_9 \rightarrow S_{12}$	T1185→T1134	0.222
$S_1 \rightarrow S_3$	Attacker→T1592	0.210	$S_4 \rightarrow S_{10}$	T1595→T1557	0.261	$S_9 \rightarrow S_{13}$	T1185→T1528	0.240
$S_1 \rightarrow S_4$	Attacker→T1592	0.210	$S_5 \rightarrow S_6$	T1217→T1040	0.140	$S_{10} \rightarrow S_{11}$	T1557→T1027	0.259
$S_1 \rightarrow S_5$	Attacker→T1592	0.339	$S_5 \rightarrow S_7$	T1217→T1548	0.148	$S_{10} \rightarrow S_{12}$	T1557→T1134	0.281
$S_2 \rightarrow S_6$	T1082→T1040	0.147	$S_5 \rightarrow S_8$	T1217→T1550	0.164	$S_{10} \rightarrow S_{13}$	T1557→T1528	0.235
$S_2 \rightarrow S_7$	T1082→T1548	0.155	$S_5 \rightarrow S_9$	T1217→T1185	0.184	$S_{11} \rightarrow S_{14}$	T1027→T1005	0.366
$S_2 \rightarrow S_8$	T1082→T1550	0.172	$S_5 \rightarrow S_{10}$	T1217→T1557	0.196	$S_{11} \rightarrow S_{15}$	T1027→T1552	0.324
$S_2 \rightarrow S_9$	T1082→T1185	0.193	$S_6 \rightarrow S_{11}$	T1040→T1027	0.306	$S_{12} \rightarrow S_{14}$	T1134→T1005	0.385
$S_2 \rightarrow S_{10}$	T1082→T1557	0.206	$S_6 \rightarrow S_{12}$	T1040→T1134	0.257	$S_{12} \rightarrow S_{15}$	T1134→T1552	0.341
$S_3 \rightarrow S_6$	T1592→T1040	0.150	$S_6 \rightarrow S_{13}$	T1040→T1528	0.278	$S_{13} \rightarrow S_{14}$	T1528→T1005	0.377

(Continued)

Table 12 (continued)

State transition	Attack transition	Transition probability	State transition	Attack transition	Transition probability	State transition	Attack transition	Transition probability
$S_3 \rightarrow S_7$	T1592→T1548	0.188	$S_7 \rightarrow S_{11}$	T1548→T1027	0.276	$S_{13} \rightarrow S_{15}$	T1528→T1552	0.333
$S_3 \rightarrow S_8$	T1592→T1550	0.175	$S_7 \rightarrow S_{12}$	T1548→T1134	0.234	$S_{14} \rightarrow S_{16}$	T1005→T1498	0.303
$S_3 \rightarrow S_9$	T1592→T1185	0.197	$S_7 \rightarrow S_{13}$	T1548→T1528	0.253	$S_{14} \rightarrow S_{17}$	T1005→T1499	0.303
$S_3 \rightarrow S_{10}$	T1592→T1557	0.209	$S_8 \rightarrow S_{11}$	T1550→T1027	0.272	$S_{15} \rightarrow S_{16}$	T1552→T1498	0.317
$S_4 \rightarrow S_6$	T1595→T1040	0.186	$S_8 \rightarrow S_{12}$	T1550→T1134	0.228	$S_{15} \rightarrow S_{17}$	T1552→T1499	0.317
$S_4 \rightarrow S_7$	T1595→T1548	0.197	$S_8 \rightarrow S_{13}$	T1550→T1528	0.250			
$S_4 \rightarrow S_8$	T1595→T1550	0.218	$S_9 \rightarrow S_{11}$	T1185→T1027	0.263			

$$P = \begin{bmatrix}
0 & 0.242 & 0.210 & 0.210 & 0.339 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.126 & 0 & 0 & 0 & 0.147 & 0.155 & 0.172 & 0.193 & 0.206 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.111 & 0 & 0 & 0.150 & 0.158 & 0.175 & 0.197 & 0.209 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.138 & 0 & 0.186 & 0.197 & 0.218 & 0.245 & 0.261 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.168 & 0.140 & 0.148 & 0.164 & 0.184 & 0.196 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.224 & 0 & 0 & 0 & 0 & 0.306 & 0.257 & 0.278 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.234 & 0 & 0 & 0 & 0.278 & 0.234 & 0.253 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.253 & 0 & 0 & 0.272 & 0.228 & 0.250 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.275 & 0 & 0.263 & 0.222 & 0.240 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.288 & 0.259 & 0.218 & 0.235 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.310 & 0 & 0 & 0.366 & 0.324 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.274 & 0 & 0.385 & 0.341 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.290 & 0.377 & 0.333 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.394 & 0 & 0.303 & 0.303 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.365 & 0.317 & 0.317 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

$$Q = \begin{bmatrix}
0 & 0.242 & 0.210 & 0.210 & 0.339 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.126 & 0 & 0 & 0 & 0.147 & 0.155 & 0.172 & 0.193 & 0.206 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.111 & 0 & 0 & 0.150 & 0.158 & 0.175 & 0.197 & 0.209 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.138 & 0 & 0.186 & 0.197 & 0.218 & 0.245 & 0.261 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.168 & 0.140 & 0.148 & 0.164 & 0.184 & 0.196 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.224 & 0 & 0 & 0 & 0 & 0.306 & 0.257 & 0.278 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.234 & 0 & 0 & 0 & 0.278 & 0.234 & 0.253 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.253 & 0 & 0 & 0.272 & 0.228 & 0.250 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.275 & 0 & 0.263 & 0.222 & 0.240 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.288 & 0.259 & 0.218 & 0.235 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.310 & 0 & 0 & 0.366 & 0.324 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.274 & 0 & 0.385 & 0.341 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.290 & 0.377 & 0.333 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.394 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.365
\end{bmatrix}$$

$$N = \begin{bmatrix}
1.000 & 0.277 & 0.236 & 0.244 & 0.407 & 0.230 & 0.246 & 0.280 & 0.324 & 0.351 & 0.566 & 0.453 & 0.502 & 0.942 & 0.795 \\
0 & 1.144 & 0 & 0 & 0 & 0.217 & 0.232 & 0.263 & 0.305 & 0.331 & 0.534 & 0.427 & 0.473 & 0.887 & 0.749 \\
0 & 0 & 1.125 & 0 & 0 & 0.217 & 0.232 & 0.264 & 0.306 & 0.330 & 0.534 & 0.427 & 0.473 & 0.888 & 0.750 \\
0 & 0 & 0 & 1.16 & 0 & 0.278 & 0.298 & 0.339 & 0.392 & 0.425 & 0.686 & 0.548 & 0.608 & 1.141 & 0.963 \\
0 & 0 & 0 & 0 & 1.202 & 0.217 & 0.232 & 0.264 & 0.305 & 0.331 & 0.534 & 0.427 & 0.473 & 0.888 & 0.750 \\
0 & 0 & 0 & 0 & 0 & 1.289 & 0 & 0 & 0 & 0 & 0.571 & 0.456 & 0.505 & 0.949 & 0.801 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.305 & 0 & 0 & 0 & 0.526 & 0.421 & 0.465 & 0.874 & 0.738 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.339 & 0 & 0 & 0.528 & 0.420 & 0.471 & 0.879 & 0.742 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.379 & 0 & 0.526 & 0.422 & 0.466 & 0.876 & 0.739 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.404 & 0.527 & 0.422 & 0.466 & 0.876 & 0.739 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.449 & 0 & 0 & 0.875 & 0.739 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.377 & 0 & 0.875 & 0.740 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.408 & 0.876 & 0.739 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.650 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.575
\end{bmatrix}$$

TEPPA first links the attack techniques into the AG, constructs the AMC based on the AG, and then maps the AMC to the EG. Fig. 10 shows the AG and the mapped attack technique evolution event graph, where the state transition probabilities on the edges have been normalized.

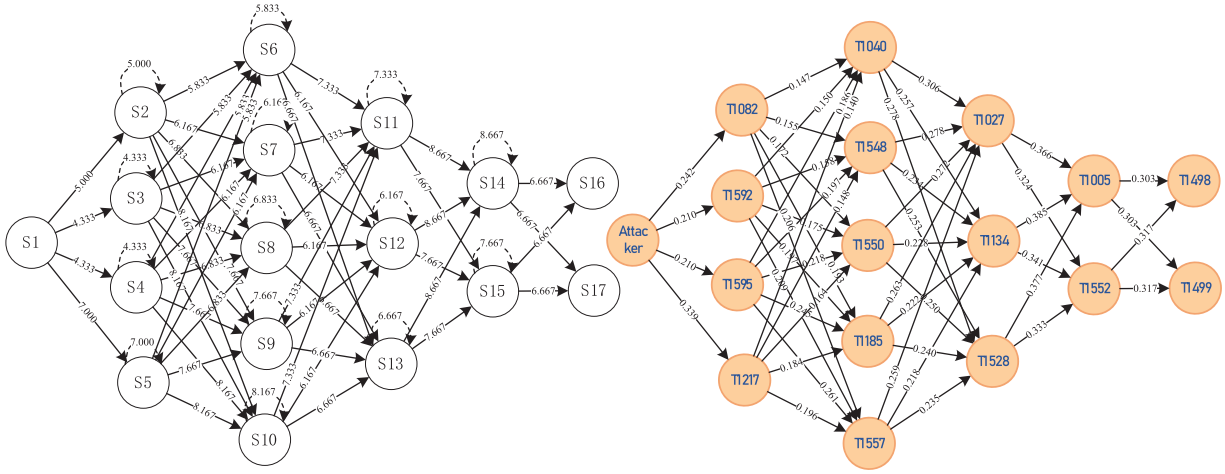

Figure 10: Mapping of attack graph to attack technique evolution event graph

In Fig. 11, the red circles indicate the attack techniques with the highest probability of being used to compromise each device. They are connected by red lines to form the attack technique evolution path with the highest probability. Finally, we integrate the prediction results from the macro and micro-layer, which enables the mapping of the attacked devices to the attack techniques. Security personnel can visualize the most likely attack paths and techniques attackers use to protect critical devices and prevent specific attack techniques better.

A device can be attacked by more than one attack technique, so when the probabilities of all possible attack techniques are summed, the higher the value, the higher the probability of the device being attacked. We regard this probability as the hazard degree of the device and determine the protection sequence of the device according to the hazard degree. In summary, the protection sequence of the device in the threat propagation path can be predicted based on the matrix N. As seen in Fig. 12, FW, Router, WS_1, and AS are the devices in the threat propagation path predicted by TPPPA. S_2–S_{15}, respectively, correspond to an attack technique, clustered according to the attacked device FW, Router,

WS_1, and AS. The bar chart shows the number of visits to each attack technique, which represents the use probability of it. The line in the graph shows the sum of the use probabilities of all attack techniques for each device, i.e., the risk degree of the device.

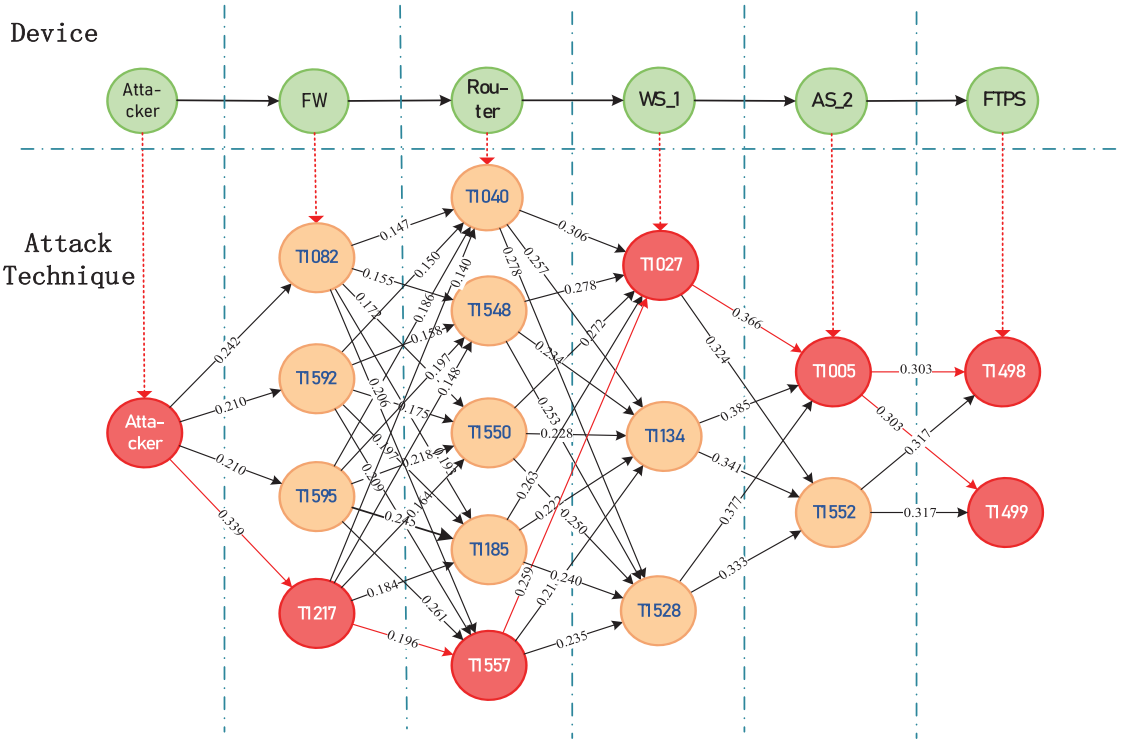

Figure 11: TL-TPM combines macro and micro-layer

Figure 12: Expected number distribution of using each attack technique

The higher the risk degree of the device, the higher the priority to protect it. Therefore, from the line in Fig. 12, we can see that the sequence of device protection in the threat propagation path predicted by AEPPA is: AS>WS_1>Router>FW. Meanwhile, the attack technique T1005, represented

by the state node S_{14}, has the highest number of node visits. It indicates that the attacker will most likely use T1005, so AS should be strengthened to defend T1005.

4.3 Contrast Analysis

In this section, to illustrate the effectiveness of TL-TPM, this paper compares it with Hu et al.'s [10] model. Specifically, TL-TPM compares the prediction results of the device repair sequence and threat propagation path, and the time complexity. Finally, it compares with several previous models in a comprehensive way.

1. Prediction of Device Repair Sequence

We use Hu Hao's method to obtain his device repair sequence for this experiment scene. The topology of the experiment scenario is shown in Fig. 13. Similarly, his method needs to derive the state transition probability matrices *P'* and *N'*, and the values in the first row of the matrix N' are used to determine the sequence of repair. The matrices *P'* and *N'* are shown below.

To illustrate the effectiveness and superiority of TL-TPM, we compare the predicted outcomes of device repair sequences of TL-TPM with Hu Hao's method. Table 13 illustrates the device repair sequences, and it can be observed that Hu Hao's method indicates that DS should be prioritized for repair when adopting network security measures, but TL-TPM indicates that AS should be prioritized for repair.

$$P' = \begin{bmatrix} 0 & 1.000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.109 & 0.641 & 0 & 0 & 0 & 0.250 & 0 & 0 & 0 \\ 0 & 0 & 0.418 & 0.163 & 0.418 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.280 & 0.719 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.719 & 0 & 0 & 0 & 0 & 0.280 \\ 0 & 0 & 0 & 0 & 0 & 0.333 & 0.333 & 0.333 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.238 & 0 & 0.524 & 0.238 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.500 & 0.500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.688 & 0.312 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$N' = \begin{bmatrix} 1.000 & 1.122 & 1.236 & 0.280 & 2.555 & 0.421 & 0.368 & 0.280 & 0.617 \\ 0 & 1.122 & 1.236 & 0.280 & 2.555 & 0.421 & 0.368 & 0.280 & 0.617 \\ 0 & 0 & 1.718 & 0.389 & 3.551 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.389 & 3.554 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.559 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.499 & 1.31 & 0.999 & 2.201 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.312 & 0 & 2.204 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.312 & 2 & 2.204 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.205 \end{bmatrix}$$

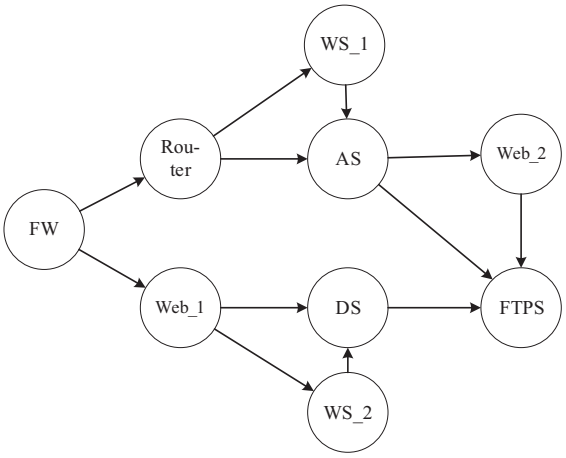

Figure 13: The topology of the experiment scenario

Table 13: The sequence of device repair in the threat propagation path

Hu Hao's method	DS>Web_1>FW>Web_2>Router>AS>WS_1=WS_2
TL-TPM	AS>Router>Web_2>WS_1>DS>FW>Web_1>WS_2

For this discrepancy, we analyze the effect of device node repair. Repairing a device node, i.e., deleting it and all edges associated with it in the topology graph, and then counting the number of remaining attack paths, the results are shown in Fig. 14. And from Fig. 13, it can be found that there are six attack paths that can attack FTPS. It is clear that when priority is given to protecting AS, i.e., the device node is removed from the graph, and the remaining attack paths are two. While the DS is removed, the remaining attack paths are four. Therefore, the result of TL-TPM is more scientific and accurate. If the device nodes are repaired sequentially according to the repair sequence in Table 13, it can be seen from Fig. 15 that both Hu Hao's method and TL-TPM leave only two attack paths after repairing the device for the third time, and leave no attack paths after repairing the device for the fourth time. But TL-TPM overall outperforms Hu Hao's approach by intercepting more attack paths earlier.

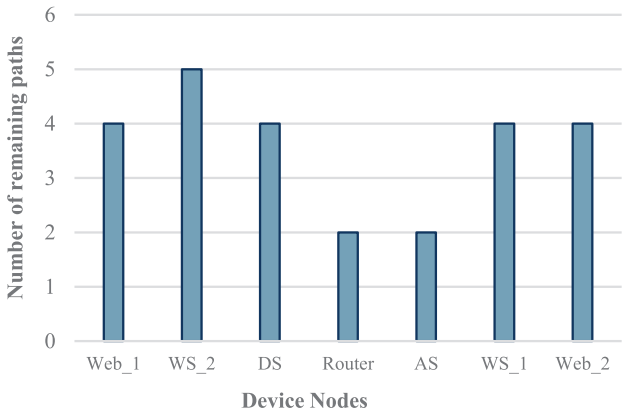

Figure 14: The number of remaining paths after node repair

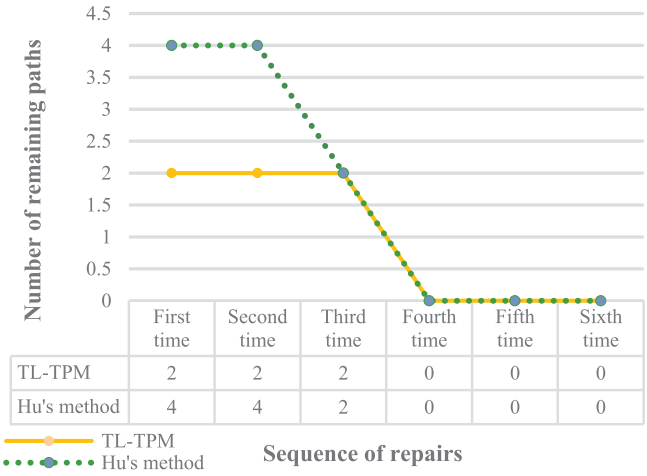

Figure 15: The remaining attack paths after repairing devices in sequence

2. Prediction of Threat Propagation Path

Next, we compare the threat propagation path predicted by TL-TPM and Hu Hao's method. Hu Hao's model first obtains the state transition probability of each device node according to P', then the state transition probabilities of the device nodes in each attack path are cumulatively multiplied to calculate the probability of success in compromising the core asset along that path. The path with the highest probability of success is used as the final predicted threat propagation path. The lengths of all threat propagation paths and their success probabilities are shown in Table 14.

Table 14: Threat propagation path lengths and their probability distributions

Route	Threat propagation path	Route length	Cumulative success probability
$Route_1$	Attacker→FW→Router→WS_1→AS→FTPS	5	0.0433
$Route_2$	Attacker→FW→Router→WS_1→AS→ Web_2 →FTPS	6	0.0137
$Route_3$	Attacker→FW→Web_1→WS_2→DS→FTPS	5	0.0277
$Route_4$	Attacker→FW→Web_1→DS→FTPS	4	0.0553

The results in the Table 14 show that $Route_4$ has the highest probability of success. Therefore, the path predicted by Hu Hao's algorithm is $Route_4$. And as seen from the previous section, the path predicted by TL-TPM is $Route_1$, which differs significantly from the path $Route_4$ indicated by Hu Hao. This is because Hu Hao's method multiplied cumulatively the transition probabilities between all devices in the path and simply chose the path with the highest cumulative success probability value, not considering that the attacker penetrated gradually. When the attacker is faced with two attackable devices, he always selects the device that is more favorable to him, i.e., the one with the higher risk degree, to attack. As shown in Table 14, although $Route_4$ has a higher cumulative success probability than $Route_1$, the risk degree of Router in $Route_1$ is higher than that of Web_1 in $Route_4$. As a result,

the attacker is more likely to choose Router to attack and follow the $Route_1$. Overall, TL-TPM takes a comprehensive view from the attacker's point to reflect the actual situation more accurately.

3. Comparison of Time Complexity

Then, we compare the time complexity of TL-TPM with that of Hu Hao's model. TL-TPM includes two layers, each containing one main algorithm.

Firstly, the time complexity of TPPPA in the macro-layer is analyzed. According to the algorithm logic, assuming that there are n devices between the initial device *ind* and the device *target* where the core asset is located. And the average number of adjacent devices at the next layer for each device is m. Then a total of $(n-1)m$ devices need to be calculated for the threat degree from the *ind* to the *target*. So, the time complexity of execution from *ind* to *target* is $O((n-1)m)$. Because m is constant, the time complexity of the algorithm is O(n).

Secondly, the time complexity of AEPPA in the micro-layer is analyzed. Executing AEPPA is based on the result of TPPPA. Assuming a total of n attack techniques are extracted from the result, calculating their state transition probabilities requires the generation of two matrices with a time complexity of $O(n^2)$. Therefore, the time complexity of TL-TPM to obtain the final prediction result is $O(n^2) + O(n)$, i.e., $O(n^2)$, while the time complexity of Hao Hu's model is $O(n^3)$. As a result, TL-TPM is superior in terms of time complexity.

4. Comparison of Other Prediction Models

Comparing TL-TPM with other attack prediction models, the results in Table 15 show that TL-TPM is more advanced with considering both macro and micro-layers to predict threat development. It considers the threat impact elements (attack success probability, threat degree) and avoids path redundancy. Furthermore, only this paper's research has the capability of predicting the threat propagation path while correlating the attacked devices with their respective threat elements, broadening the range of predictions. Moreover, TL-TPM can accurately predict the attack techniques, not only letting security personnel know which devices should be protected in priority but also which attack techniques should be strengthened against.

Table 15: Comparison of attack prediction models

Model	Macro	Micro	No redundant paths	Take threat impact elements into account	Correlate the threat elements
Gong et al. [5]	√	×	×	×	×
Wu et al. [6]	√	√	×	×	×
Hu et al. [10]	√	×	×	√	×
Yuan et al. [11]	√	×	×	×	×
Wang et al. [13]	×	√	×	√	×
Zhang et al. [14]	√	√	×	×	×
Sun et al. [15]	√	√	√	√	×
TL-TPM	√	√	√	√	√

5 Conclusion

Unlike most previous works that predict the attack based on only one layer, this paper proposes a two-layer model TL-TPM that predicts the development trend of threat events from both macro and micro-layers. The macro-layer proposes the threat propagation path prediction algorithm TPPPA based on the knowledge graph. TPPPA measures the device threat degree by combining system topology and attack success probability. Based on the device threat degree, it predicts the devices under attack, then links them sequentially into threat propagation path and correlates each device with relevant threat elements, which provides decision support for defense response. The micro-layer proposes the attack evolution path prediction algorithm AEPPA based on the event graph. AEPPA combines the prediction results of the macro-layer with the temporal characteristics of the attack behaviors and innovatively maps the attack graph to the event graph using the absorbing Markov chain as a bridge, which accurately portrays the evolution of the attack techniques used in threat events. Finally, the macro-layer and micro-layer prediction results are integrated to visualize the external path and internal logic of threat event development, enabling security personnel to quickly grasp the threat status of system devices and focus on defense.

However, TL-TPM does not consider zero-day vulnerabilities when predicting threats, and the current algorithms and inference rules only work with known vulnerabilities. For future work, we will use the relationship paths linking attacker entities to target entities in the knowledge graph as features and construct attack samples using historical attack data for the given system. Then, we use machine learning to learn the path features in the attack samples to distinguish the zero-day vulnerabilities from the known vulnerabilities. Meanwhile, TL-TPL does not consider the vulnerability lifecycle, which may affect the calculation of the attack success probability. As a result, we will take the vulnerability lifecycle into account, quantitatively analyze the change in vulnerability exploitability over time, optimizing the calculation of the state transition matrix.

Acknowledgement: The authors would like to thank the reviewers for the correct and concise recommendations that help present the materials better.

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

Author Contributions: The authors confirm contribution to the paper as follows: methodology: Shuqin Zhang; conceptualization: Shuqin Zhang, Xinyu Su; investigation: Yunfei Han; data curation: Peiyu Shi; analysis and interpretation of results: Yunfei Han, Tianhui Du; validation: 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 and data can be obtained by contacting the corresponding author.

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

References

[1] J. Zhao, Q. Yan, J. Li, M. Shao, Z. He *et al.,* "TIMiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data," *Computers & Security*, vol. 95, pp. 101867, 2020.

[2] J. Zhao, Q. Yan, X. Liu, B. H. Li and G. Zuo, "Cyber threat intelligence modeling based on heterogeneous graph convolutional network," in *Proc. of the 23rd Int. Symp. on Research in Attacks, Intrusions and Defenses (RAID 2020)*, San Sebastian, Spain, pp. 241–256, 2020.
[3] G. X. Xu, M. X. Hu and C. Ma, "Secure and smart autonomous multi-robot systems for opinion spammer detection," *Information Sciences*, vol. 576, pp. 681–693, 2021.
[4] J. Zhao, M. L. Shao, H. Wang, X. M. Yu, B. Li *et al.,* "Cyber threat prediction using dynamic heterogeneous graph learning," *Knowledge-Based Systems*, vol. 240, pp. 108086, 2022.
[5] 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 (In Chinese).
[6] 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.
[7] 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.*, New York, NY, USA: Association for Computing Machinery, pp. 1–4, 2015.
[8] 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.
[9] J. Zhao, X. D. Liu, Q. B. Yan, B. Li, M. L. Shao *et al.,* "Automatically predicting cyber attack preference with attributed heterogeneous attention networks and transductive learning," *Computers & Security*, vol. 102, pp. 102152, 2021.
[10] H. Hu, Y. L. Liu, H. Q. Zhang, Y. J. Yang and R. G. Ye, "Route prediction method for network intrusion using absorbing Markov chain," *Journal of Computer Research and Development*, vol. 55, pp. 831–845, 2018.
[11] 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.
[12] 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.
[13] S. Wang, G. M. Tang and G. Kou, "Attack path prediction method based on causal knowledge net," *Journal on Communications*, vol. 37, pp. 188–198, 2016.
[14] 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.
[15] 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.
[16] NIST, "Common platform enumeration," [Online]. Available: https://nvd.nist.gov/Products/CPE (accessed on 21/03/2023)
[17] MITRE, "Common vulnerabilities and exposure," [Online]. Available: https://cve.mitre.org (accessed on 21/03/2023)
[18] NIST, "National vulnerability databased," [Online]. Available: https://nvd.nist.gov (accessed on 21/03/2023)
[19] MITRE, "Common weakness enumeration," [Online]. Available: https://cwe.mitre.org/ (accessed on 21/03/2023)
[20] MITRE, "Common attack pattern enumeration and classification," [Online]. Available: https://capec.mitre.org (accessed on 21/03/2023)
[21] MITRE, "ATT&CK matrix for enterprise," [Online]. Available: https://attack.mitre.org/ (accessed on 21/03/2023)
[22] MITRE, "D3FEND," [Online]. Available: https://d3fend.mitre.org (accessed on 21/03/2023)
[23] FIRST, "Common vulnerability scoring system," [Online]. Available: https://www.first.org/cvss/ (accessed on 21/03/2023)
[24] S. E. Wang, C. X. Liu and X. S. Liu, "A method of 5G network security risk assessment based on attack graph," *Computer Applications and Software*, vol. 40, pp. 289−296+335, 2023.

[25] V. V. Kovtun, I. Izonin and M. Greguš, "The functional safety assessment of cyber-physical system operation process described by Markov chain," *Scientific Reports*, vol. 12, pp. 7089, 2022.
[26] H. Y. Kang and M. L. Long, "Research on network attack analysis method based on attack graph of absorbing Markov chain," *Journal on Communications*, vol. 44, pp. 122–135, 2023.