Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 5, pp. 3491–3499, DOI 10.12694/scpe.v25i5.3035

DESIGN OF 0-DAY VULNERABILITY MONITORING AND DEFENSE ARCHITECTURE BASED ON ARTIFICIAL INTELLIGENCE TECHNOLOGY

JIAN HU*, ZHENHONG ZHANG†, FEILU HANG ‡ AND LINJIANG XIE§

Abstract. In response to the difficulty in detecting attacks caused by the unknown nature of 0-day vulnerabilities, the author proposes a knowledge graph based 0-day attack path prediction method. By extracting concepts and entities related to attacks from existing research on the ontology of network security and network security databases, a network defense knowledge graph is constructed to extract discrete security data such as threats, vulnerabilities, and assets into interrelated security knowledge. Using a knowledge graph reasoning method based on path sorting algorithm to explore possible 0-day attacks in the target system. Experimental results have shown that the proposed method can rely on the knowledge system provided by the knowledge graph to provide comprehensive knowledge support for attack prediction, reduce the dependence of prediction analysis on expert models, and effectively overcome the adverse effects of unknown 0-day vulnerabilities on prediction analysis. It improves the accuracy of 0-day attack prediction and utilizes the path sorting algorithm to infer based on the explicit feature of graph structure, being able to effectively backtrack the reasons behind the formation of reasoning results, this to some extent improves the interpretability of attack prediction analysis results.

Key words: 0day attack, Attack path prediction, Artificial intelligence, Defense architecture

1. Introduction. With the continuous development of computer and communication technologies, network security has increasingly become an important issue affecting network performance and data security. The proliferation of network attacks and viruses poses a serious threat to network and application systems. For enterprise level users, whenever they encounter these threats, they often cause data damage, system abnormalities, network paralysis, information theft, decreased work efficiency, and significant direct or indirect economic losses. Since 2006, the US Security Training and Research Institute (SANS) has included zero day attacks as one of the top 20 global internet security threats annually [1]. Enterprises that have not yet patched zero day vulnerabilities have become the preferred targets for hackers, and such attacks are developing rapidly. Zero day attack refers to an attack launched by malicious software that exploits certain vulnerabilities in the operating system or application software that are not known to developers or have not been patched in a timely manner. Those vulnerabilities that are not known to developers or have not been patched in a timely manner are also known as "zero day vulnerabilities". As a type of attack, the difference between zero day attacks and traditional hacker attacks is that the target of zero day attacks is some potential unknown or publicly disclosed but not patched vulnerabilities. According to authoritative institutions, there may be 4-5 coding vulnerabilities in every 1000 lines of code in operating systems and applications currently in use[2]. With the continuous emergence of various computer vulnerabilities, the situation of zero day attacks is also constantly changing: From a single point to a desktop type, then gradually transitioning to a network type, and even currently there is a trend towards a full network type development. In this complex form, defense against zero day attacks is also constantly evolving. Traditionally, regular updates of system patches, firewalls, intrusion detection systems, and antivirus software are commonly used to protect critical business and IT infrastructure. These systems provide good first level protection, but still cannot avoid zero day vulnerability attacks. Faced with the increasing

*Network Security Management Center of Information Center of Yunnan Power Grid Co., LTD., Kunming, Yunnan, China,650000 (Corresponding author, hjiang2023@126.com)
†Network Security Management Center of Information Center of Yunnan Power Grid Co., LTD., Kunming, Yunnan, China, 650000
‡Information Security Operation and Maintenance Center of Information Center of Yunnan Power Grid Co., LTD., Kunming, Yunnan, China, 650000
§Information Security Operation and Maintenance Center of Information Center of Yunnan Power Grid Co., LTD., Kunming, Yunnan, China, 650000

zero day threat, both system, network, and security vendors are loudly calling for the importance of real-time updates [3,4]. For manufacturers, the purpose of instant updates is to respond to the increasingly short attack time gap of hackers. However, users often lack manpower, resources, and time to effectively perform the work of instant updates and repairs.

The main reason why many enterprises are unable to immediately complete the repair work of system or software and hardware device vulnerabilities is that they do not have time to discover the vulnerabilities, do not further evaluate and diagnose the vulnerabilities, and are unable to patch and update all computers or endpoint devices. Another more important reason is the inability to conduct compatibility testing on patches, which often leads to system instability and even crashes.

Attack prediction technology is the key to research on 0day attack detection. However, research on 0-day attack prediction generally relies on hypothetical conditions, attack models constructed by expert knowledge, and the pre - and post attack dependencies in the same attack path to address the impact of unknown 0-day vulnerabilities, there are three shortcomings in this process: Firstly, the conditional assumption lacks effective constraints, which can easily lead to a large scale of prediction results for 0-day attacks, reducing the significance of prediction; The second is the attack model constructed by expert knowledge, which is easily constrained by the subjective knowledge of experts; Thirdly, the prediction method is difficult to apply when the known attack path is incomplete. In response to the above shortcomings, the author proposes a 0-day attack path prediction method based on a knowledge graph [5]. By using a network defense knowledge graph, attack related threats, assets, vulnerabilities, and other data are fused into a security knowledge base that is interrelated and covers a wide range of knowledge. Based on the integrated vulnerability data, attack intent, and other knowledge, reasonable constraints are imposed on the assumptions of unknown attributes of 0-day vulnerabilities; Secondly, using path sorting algorithms, the relationship path between the attacker entity and the target entity in the knowledge graph is used as a feature to predict the 0-day attack from a more comprehensive perspective, overcoming the limitations of expert knowledge construction models [6]; Finally, using historical attack data as samples, a logistic binary classifier is designed and trained to implement single step attack prediction, thereby breaking away from dependence on known attack paths. By reusing the single step attack probability output by the logistic binary classifier, the comprehensive utilization rate of the attack path is calculated to predict the 0-day attack path most likely to be exploited by the attacker against the target asset, thereby supporting defense decisions [7,8].

2. Methods.

2.1. Preparatory knowledge. In order to make the expression clear and accurate, the relevant concepts in the text are defined as follows.

Definition of Network Defense Knowledge Graph (CKG). CKG is represented by triplets (CSO, FACT, T), where CSO=(C, R, P) is the network security ontology, C is the class set, R is the relationship type set, P is the attribute type set, FACT is the set of data knowledge represented in RDF (resourcedescriptionframework) triplet format, T is the set of type dependency relationships between classes in CSO and entity objects in FACT [9].

Define a 20day vulnerability. 0day vulnerability refers to a general term for system vulnerabilities that have not been discovered by security vendors but may be mastered by hacker organizations. In order to make the research more targeted, the following 0day vulnerabilities only refer to technical vulnerabilities that have not been discovered by security vendors.

Define a 30 day attack. A 0-day attack refers to a single step attack initiated by an attacker using a known 0-day vulnerability, denoted as a^0. Relatively, known attack a^k is a single step attack initiated by an attacker exploiting a known vulnerability.

Define 40 day attack path zap. Zap refers to an acyclic attack sequence consisting of a set of single step attacks with 0 day dependencies, represented by (A, E), where A is the set of single step attacks and E is the directed edge set of linked single step attacks [10].

Define 50 day attack graph ZAG. ZAG refers to an attack graph containing 0-day attacks, represented as (A, Priv, L, Prob), among them, $A = \{a^0\} \cup \{a^k\}$ is a single step attack set consisting of 0 day attacks and known attacks. Single step attack a is represented as a binary (host, Vul), where the host is the target device, Vul is the vulnerability exploited, and Priv is the pre - and post permission set for single step attacks,

Table 2.1: Main Symbols and Their Description

symbol	describe	symbol	describe
CKG	Network Defense Knowledge Graph	vul	Vulnerabilities exploited by single step attacks
CSO	Network Security Ontology	rp	Relationship Path
zap	0day attack path	c	Classes in ontology
ZAG	0day attack graph	r	Relationship types in ontology
A	Single Step Attack Set	Domain(r)	Definition domain of relationship types
a	Single step attack	Range(r)	The range of values for relationship types
Priv	Node permission set	s	Source entity
L	Directed Link Set between Single Step Attack and Permissions	d	Target entity
E	Directed Edge Set Between Single Step Attacks in Attack Paths	h	Relationship Path Eigenvalues
Prob	Probability set of single step attack occurrence	H	Relationship path feature vector
host	Devices that have been attacked	θ	Relationship path weight vector

L={AxPriv}U{PrivxA} is the link between a single step attack and permissions, representing the pre - and post relationship between them. Prob is the set of probabilities of a single step attack occurring.

Define 6 relationship paths (rp, relationpath). rp refers to a sequence composed of a set of relationship types in a knowledge graph, written as $rp : c_0 \xrightarrow{r_1} c_1 \xrightarrow{r_2} ... \xrightarrow{r_l} c_l, r \in R$. Among them, $c_l \equiv Range(rp), l = |rp|$ represents the length of the relationship path, which is the total number of relationship types included in the relationship path, $0 \leqslant i \leqslant 1$.

The relationship between knowledge graph and attack graph is as follows: CKG is the input of attack prediction algorithm, serving as a knowledge base to provide the necessary knowledge for attack prediction; 0day attack graph ZAG is a graphical representation of attack prediction results. The difference between relational paths and attack paths is as follows: rp is used as a feature in the logistic regression model in path sorting algorithms to perform attack prediction; The 0-day attack path zap is a prediction result of the extracted attack path based on the 0-day attack graph, combined with the probability of multi-step attacks occurring. The main symbols that appear are described in Table 2.1 [11,12].

2.2. Network Defense Knowledge Graph. Knowledge graph is an effective technical method that uses graph models to describe knowledge and model the relationships between things. Applying this technology to the field of network security and constructing a network defense knowledge graph can integrate heterogeneous and fragmented network security data into a unified and interrelated security knowledge format, providing support for attack prediction and testing the required knowledge. The 0-day attack graph ZAG is a graphical representation of the attack prediction results. The difference between relational paths and attack paths is as follows: RP is used as a logistic regression model in path sorting algorithms to perform attack prediction, which is beneficial for implementing targeted defense. The following is a detailed introduction to the construction of the network defense knowledge graph architecture and the design of the network security ontology.

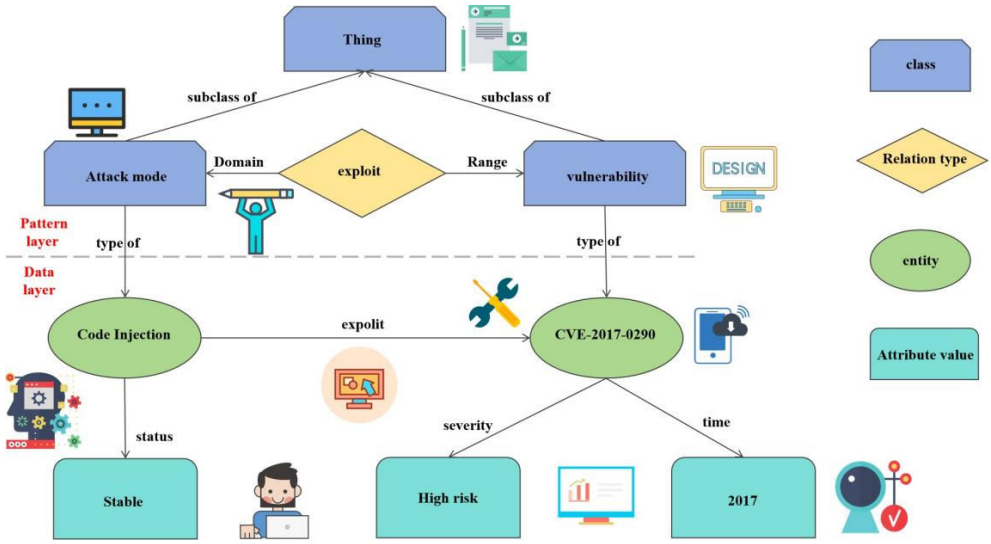

Fig. 2.1: Example of the Relationship between Pattern Layer and Data Layer

(1) Architecture construction. According to Definition 1, the network defense knowledge graph can be divided into two parts: The pattern layer and the data layer. Among them, the pattern layer is the core of the knowledge graph, and the basic concept system of network defense is defined by the Cybersecurity ontology (CSO), providing pattern definitions for modeling data layer knowledge[13]. The data layer is the main body of the knowledge graph, which is a collection of data knowledge obtained through knowledge extraction, knowledge fusion, and other steps, modeled under the pattern definition. Data knowledge is represented in the form of RDF triplets as (subject, predicate, object). The example of a two-layer relationship is shown in Figure 2.1. The pattern layer of this example defines attack pattern classes, vulnerability classes, and the relationship type exploit with the two as defined cities and value ranges, respectively. The data layer models data knowledge based on this pattern (CVE-2017-0290, CodeInjection, exploit), indicating that code injection can exploit vulnerability CVE-2017-0290.

According to the hierarchical structure of the knowledge graph, there are mainly two methods for its construction: top-down and bottom-up. Due to the mature research on the conceptual system in the field of network security, the network defense knowledge graph is suitable for a top-down knowledge graph construction method, which first constructs a pattern layer based on the network security ontology, and then integrates multiple knowledge extraction and fusion technologies based on the pattern layer to extract and model data knowledge from heterogeneous data sources and construct a data layer [14,15].

(2) Ontology design. From the top-down sequence of knowledge graph construction, it can be seen that CSO is the key to determining the quality of network defense knowledge graph. The author uses ontology integration to construct CSO, integrating existing mature network security ontologies into a unified ontology, drawing on current research achievements in this field, and achieving complementary advantages among different achievements[16]. Due to NSSEKB_ 0 (ontology of network security situa tion element knowledge base) systematically sorts out the network security knowledge system from three levels: domain ontology, application ontology, and atomic ontology, and has good knowledge completeness. AFACSDO (asset infrastructure security domain ontology) reuses multiple representative network security ontologies, which is the latest achievement in network security ontology research and has good timeliness. Therefore, the author selected the above two research results as integration objects to implement ontology integration.

2.3. 0-day attack path prediction. The network defense knowledge graph defines the concept of attacks as a type of relationship with attacker classes as the domain and device classes as the value domain. The inference problem of specific attack behaviors is transformed into the prediction problem of attack relationships

between attacker entities and device entities in the data layer of the knowledge graph. The path ranking algorithm (PRA) is an effective method for predicting knowledge graph links. Its prediction results not only have high accuracy, but also strong interpretability, making it easy for defenders to mine knowledge about attack causes and other factors after obtaining the prediction results, therefore, the author chooses the PRA algorithm to perform one-step attack prediction and constructs a 0-day attack graph. On this basis, by analyzing and comparing the comprehensive utilization rates of different attack paths, predict the most likely 0-day attack path that attackers are likely to utilize [17].

(1) Analysis of Unknown Properties of 0day Vulnerability. Before implementing the prediction, the basis of prediction analysis is to make assumptions and implement constraints on the missing 0day vulnerability attributes through unknown attribute analysis. The unknown attributes of the 0day vulnerability mainly include the location of the vulnerability, exploitation conditions, and impact information. Current research mainly supplements unknown 0-day vulnerability information through conditional assumptions. In this process, how to reasonably constrain the assumed vulnerability information is the key to determining the quality of prediction results. Based on the knowledge graph integration, the author proposes hypotheses about the existence, availability, and harm of 0-day vulnerabilities, and uses statistical analysis, sample training, and intention analysis to constrain the relevant hypotheses.

Existence assumption: The 0day vulnerability may exist in any component of the device. The vulnerability data provided by databases such as NVD and CNNVD show significant differences in the number of vulnerabilities exposed by different components, indicating that the existence of vulnerabilities is related to the components that serve as their carriers. Therefore, for this assumption, component features can be used to constrain and consider the 0-day vulnerability as a potential vulnerability that the component may expose in the future. By statistically analyzing the vulnerability exposure history data of different components, the possibility of 0-day vulnerabilities in different components can be quantified[18].

Availability Assumption: The 0day vulnerability may be triggered by arbitrary permissions on the target device. Permissions exist in the form of permission entities in the knowledge graph, and triggering different vulnerabilities requires varying degrees of permission. The higher the permission, the easier it is for attackers to trigger vulnerabilities. Set the attribute "tri_prob" of the relationship "trigger" in the knowledge graph to indicate the probability of permission triggering a 0-day vulnerability, and assign values based on the level of permission. The relationship "trigger" is contained in the relationship path between the attacker entity and the target device entity, so "triprob" can be reflected in the sample features by participating in the calculation of path features. In the process of training a Logistic binary classifier using attack samples, the exploitation conditions of the 0-day vulnerability can be constrained to the scenarios in the samples based on the difference between positive and negative samples. When the classifier determines that the 0-day attack relationship is valid, the permissions used to trigger the 0-day vulnerability can be determined by querying the permission entities that the attacker entity has obtained on the device entity in the knowledge graph.

Harmful assumption: The harm generated by exploiting a 0-day vulnerability can only meet the minimum requirement for attackers to achieve their attack intent on the target device.

The harm caused by vulnerability exploitation is represented in the form of entities in the knowledge graph, and at the same time, the knowledge graph integrates the attacker's attack intent by threatening entities. Due to the fact that attackers exploit vulnerabilities to launch attacks with the aim of achieving the attack intent, if the consequences of exploiting vulnerabilities are not sufficient to support the implementation of the attack intent, then the attacker is not necessary to exploit the vulnerability. When the consequences exceed the need to achieve the attack intent, the excess is not significant to the attacker, therefore, the above assumptions about using attack intent to constrain the harm caused by exploiting 0day vulnerabilities are feasible and reasonable. For example, the attacker entity APT-28 mainly engages in theft activities, and the target device entity stores confidential information. A threat entity "stealing secrets" is constructed in the knowledge graph as APT-28's attack intention against the device. If it is predicted that APT-28 has launched an attack on the device using the 0day vulnerability, the resulting consequences are constrained by the threat entity as "confidentiality damage", without further impact on integrity, availability, and other aspects.

(2) Attack path prediction. Using the historical attack data of a given system to construct attack samples, a classifier LC is trained to predict whether an attack has occurred. Based on this, 0 day attacks and known

attacks are distinguished from the positive attack samples, and 0 day attack samples are constructed. The classifier LCz is trained to determine whether a single step attack occurred as an 0 day attack. After completing the attack prediction, utilize the query function of the graph database to mine the vulnerability and pre - and post conditions of attack exploitation based on the starting and ending entities and relationship paths, construct a single step attack, and generate a 0-day attack graph. Based on the 0-day attack graph, with the attacker's initial permissions as the starting point and the target permissions as the endpoint, extract the 0-day attack path, and calculate the comprehensive utilization rate of different attack paths by reusing the probability of a single step attack output by the LCA classifier. Based on this, predict the most likely 0-day attack path that the attacker is likely to use. The calculation of comprehensive utilization rate is shown in Equation 2.1.

$$Exploit(zap) = \prod_{a \in zap} prob(a) \tag{2.1}$$

3. Results and Analysis. In order to verify the effectiveness of the method, the experimental environment consists of three subnets, with firewalls deployed between the subnets to achieve access control. Among them, the web server and email server deployed in the DMZ region respectively provide external application service interfaces and internal email services: subnet 1 is the office area, where two hosts and one file server are deployed, and the file server stores enterprise confidential files; subnet 2 is the business area, where application servers are deployed to provide application business support for the web server. The distribution of vulnerabilities in the system is divided into 10 training scenarios and 1 experimental scenario, respectively, for training classifiers and implementing predictions. Among them, the classifier was trained using 9 sets of attack samples generated from training scenarios, and the remaining 1 set was used to generate a test set. The performance parameters of the classifier were tested. The experimental scenario was mainly used to generate 0-day attack maps and predict 0-day attack paths. Compared with existing research results, the advantages of this method were tested.

3.1. Sample Training. Based on the attack data of CTF teams simulating attackers on target systems in different training scenarios, relying on knowledge graphs, the successfully attacked devices are used as target entities to calculate path characteristics and construct attack positive samples $\{(H_j, y = 0)\}$. The failed and unselected devices are used as target entities to construct attack negative samples $\{(H_j, y = 0)\}$. Using the model sklearn.linearmodel in the Python 3.5 environment_LogisticRegression constructs a binary classifier and trains LCA using attack samples generated from training scenarios 1-9. On this basis, in the attack positive samples, distinguish between 0 day attacks and known attacks, construct 0 day attack positive samples and negative samples respectively, and train LCz. Use 5-fold cross validation to obtain the learning curve of the classifier, as shown in Figures 3.1 and 3.2[19].

Using the samples generated from training scenario 10 as the test set, the classifier was tested and the predicted results were compared with the actual attack situation. The accuracy of the classifier's LCA was 0.875, the recall was 0.917, and the harmonic mean F1 was 0.883. The classifier LCz's prediction results for the test set are consistent with the actual situation. It can be seen that the classifier has good recognition ability against 0-day attacks.

3.2. Experimental Results. Use the trained classifiers LCA and LCz to predict possible attacks in the target system in the experimental scenario. The predicted attack process is as follows: the attacker first utilizes remote access privileges to access firewalls_1. Initiate a 0-day attack, break through access control, and obtain remote access to the E-MailServer. Utilize the CVE-2018-18772 vulnerability in its operating system CentOS to launch a cross site request attack, obtain access to Host_1 and firewall2, and launch code injection attacks using the existing CVE-2018-12714 and CVE-2017-17156 vulnerabilities in both, through Host_1. Obtain access to File_server and access to Host2 through firewall2. At this time, the access permission can be directly used to launch a 0-day attack on File_server or Host2, obtaining user permissions for File_Server and triggering known vulnerabilities such as CVE-2018-8169. The probability of a single step attack occurring in LCA output is shown in Table 3.2.

With the ultimate goal of obtaining root_File_Server, the 0-day attack path is extracted, including zapl: a1-→ $a2 \rightarrow a3 \rightarrow a4$ and zap2: $a1 \rightarrow a2 \rightarrow a5 \rightarrow a6 \rightarrow > a7$. The comprehensive utilization rates of the two paths are calculated using Equation 2.1 to be 0.18 and 0.21, respectively. From this, it can be seen that

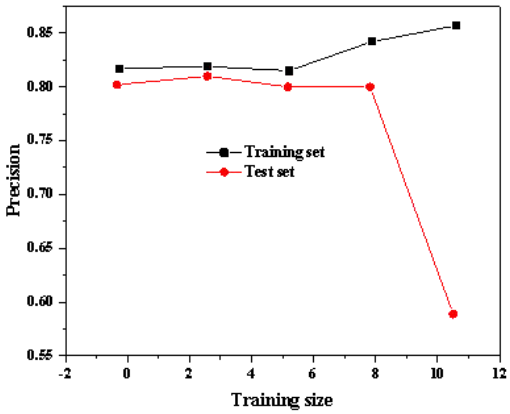

Fig. 3.1: LCA Learning Curve

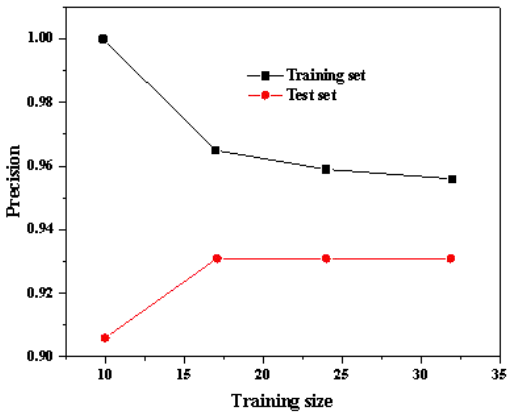

Fig. 3.2: LCZ Learning Curve

Table 3.1: Probability of attack occurrence

code	Single step attack	Probability of attack occurrence
a1	(Firewall_1 0day-001)	0.51654178
a2	(Email_Server CVE-2018-18772)	0.77063787
a3	(Host_1 CVE-2018-12714)	0.78201284
a4	(File_Server 0day-003)	0.58126118
a5	(Firewall_2 CVE-2017-17156)	0.76765849
a6	(Host_2 0day-005)	0.77269452
a7	(File_Server CVE-2018-8169)	0.9277652

Table 3.2: Comparative Analysis

method	accuracy	Convenience	Applicability	Comprehensive knowledge	Interpretability
Method A	Lower	Lower	higher	Lower	Not supported
Method B	higher	Lower	Lower	Lower	Not supported
Author's method	higher	higher	higher	higher	support

although the attack path zap2 involves 5 attacks, more than zapl's 4 attacks, its comprehensive utilization rate is higher and more likely to be exploited by the attacker.

3.3. Experimental analysis. Based on the experimental results, by comparing the author with the other two A and B methods, analyze the advantages of this method in terms of accuracy, convenience, applicability, comprehensiveness of relevant knowledge, and interpretability of prediction results, in order to verify its effectiveness. The specific comparative analysis is shown in Table 3.2.

In terms of accuracy, A's simple prediction of attack path zap1 based on the number of 0-day attacks is more likely to become the attack path chosen by the attacker. The author, by constraining the assumption of the existence of 0-day vulnerabilities (the most likely component in Web_Server to have 0-day vulnerabilities is the Apache webserver component, with a probability of 0.05), and using the trained classifier LCA to calculate the probability of an attack on Web_Server (with a probability of 0.3), determined that the attack relationship did not hold, and reasonably excluded the 0-day attack here, reducing the size of the 0-day attack prediction results, at the same time, based on the comprehensive utilization rate, Zap2 can more reasonably predict the attack path chosen by the attacker, improving the accuracy of prediction.

In terms of convenience, A and B not only need to collect relevant knowledge before implementing prediction, but also need to construct specialized 0-day attack rules as the basis for prediction analysis, which causes certain expenses. However, the author relies on network defense knowledge graph and knowledge graph inference methods to implement prediction, without the need for specialized 0-day attack rules, saving expenses and improving the convenience of the method.

In terms of applicability, method B relies on a relatively complete known attack path to implement attack inference. However, in this experimental environment, Web_server does not have a known vulnerability, and firewall_1's known vulnerability, CVE-2019-1934, requires user level permission to trigger. Therefore, it is unable to generate a known attack path under initial permission conditions, and method B is not suitable for such scenarios.

In terms of knowledge comprehensiveness, the author builds a network defense knowledge graph based on the mature conceptual knowledge in the field of network security, from three aspects: threat, asset, and vulnerability, and extracts relationship paths as features to apply to attack prediction. The prediction process not only uses vulnerability knowledge such as the existence, availability, and impact of vulnerabilities involved in A and B, but also combines knowledge of attack intent and asset types, making prediction analysis more comprehensive and improving the rationality of prediction results[20].

4. Conclusion. The author proposes a 0-day attack path prediction method based on network defense knowledge graph to address the difficulty of detecting 0-day attacks caused by the unknown nature of 0-day vulnerabilities, as well as the shortcomings of existing research in using conditional assumptions and correlation before and after attacks to overcome the impact of unknowns. By utilizing the mature knowledge of network security ontology in current research, a comprehensive network defense knowledge graph has been constructed, integrating discrete threat, vulnerability, and asset knowledge into a highly correlated knowledge system, providing comprehensive knowledge support for attack prediction. On this basis, the attack prediction problem is transformed into a link prediction problem. A path sorting algorithm with high prediction accuracy and strong interpretability of prediction results is selected to extract the relationship paths between the attacker entity and the target device entity as features, and more comprehensively predict the attack. This effectively over-

comes the influence of unknown 0-day vulnerabilities and one-sided expert knowledge, and improves prediction accuracy, and provided support for the interpretability of the predicted results. The next step is to expand the knowledge module, improve the accuracy of attack prediction, and explore the traceability problem of network attackers based on knowledge graphs in the presence of multiple attackers simultaneously.

5. Acknowledgement. Yunnan Power Grid CO., LTD. Science and Technology Project "Web Application Protection Based on RBI Remote Browser Isolation Technology" (NO.:059300KK52220011)

REFERENCES

[1] Khaoula, T., Abdelouahid, R. A., Ezzahoui, I., & Marzak, A. (2021). Architecture design of monitoring and controlling of iot-based aquaponics system powered by solar energy - sciencedirect. Procedia Computer Science, 191(33), 493-498.
[2] FAN Xue-wei,XIE Feng,WANG Xiao-wu,TANG Nan. (2023). Design of remote monitoring system for electric torque wrench based on b/s and c/s fusion architecture. Manufacturing Automation, 45(2), 175-178.
[3] Yusuf, W. (2022). The design of integrated fire spot monitoring system for industrial plantation forest using enterprise architecture approach. EDPACS: The EDP audit, control and security newsletter44(2), 66.
[4] Qian, H. (2021). Design of tunnel automatic monitoring system based on bim and iot. Journal of Physics Conference Series, 1982(1), 012073.
[5] Sangeetha, M., Thejaswini, G., Shoba, A., Gaikwad, S. S., Amretasre, R. T., & Nivedita, S. (2021). Design and development of a crop quality monitoring and classification system using iot and blockchain. Journal of Physics: Conference Series, 1964(6), 062011 (14pp).
[6] Katpatal, Y. B., & Singh, C. K. (2023). Conjunctive use of flow modelling, entropy, and gis to design the groundwater monitoring network in the complex aquifer system. International Journal of Hydrology Science and Technology, 15(1), 78-.
[7] Zhang, H., Ge, D., Yang, N., Jia, P., & Yang, Y. (2021). Study on internet of things architecture of substation online monitoring equipment. MATEC Web of Conferences, 336(5), 05024.
[8] Sun, C., Hao, H. U., Yang, Y., & Zhang, H. (2022). Prediction method of 0day attack path based on cyber defense knowledge graph. Chinese Journal of Network and Information Security, 8(1), 151-166.
[9] Lindberg, L., Vinners, B., & Lalander, C. (2022). Process efficiency in relation to enzyme pre-treatment duration in black soldier fly larvae composting. Waste Management, 137(45), 121-127.
[10] Chen, D., Yan, Q., Wu, C., & Zhao, J. (2021). Sql injection attack detection and prevention techniques using deep learning. Journal of Physics: Conference Series, 1757(1), 012055 (7pp).
[11] (2021). Elective cesarean delivery at term and its effects on respiratory distress at birth in japan: the japan environment and children's study. Health Science Reports, 4(4),46.
[12] Chen, J., Kong, Q., Sun, Z., & Liu, J. (2021). Freshness analysis based on lipidomics for farmed atlantic salmon (salmo salar l.) stored at different times. Food chemistry, 67(7),131564.
[13] Luo, S., Wang, Z., Li, X., Onchari, M. M., & Jin, S. (2021). Feed deprivation over 16 days followed by refeeding until 75 days fails to elicit full compensation of procambarus clarkii. Aquaculture, 547(34), 737490.
[14] Meng, B., Smith, W., & Durling, M. (2021). Security threat modeling and automated analysis for system design. SAE International Journal of Transportation Cybersecurity and Privacy765(1), 4.
[15] Chondamrongkul, N., Sun, J., & Warren, I. (2021). Formal security analysis for software architecture design: an expressive framework to emerging architectural styles. Science of Computer Programming, 206(27), 102631.
[16] Petrillo, A., Murino, T., Piccirillo, G., Santini, S., & Caiazzo, B. (2023). An iot-based and cloud-assisted ai-driven monitoring platform for?smart manufacturing: design architecture and experimental validation. Journal of Manufacturing Technology Management, 34(4), 507-534.
[17] Behmel, S.Damour, M.Ludwig, R.Rodriguez, M. J. (2021). Intelligent decision-support system to plan, manage and optimize water quality monitoring programs: design of a conceptual framework. Journal of Environmental Planning and Management, 64(3a4),43.
[18] Dutta, A., & Kumar, A. (2022). The imperative relationship between architecture, urban design and development and disaster management. ECS transactions35(1), 107.
[19] Bortsova, G., Cristina González-Gonzalo, Wetstein, S. C., Dubost, F., & Bruijne, M. D. (2021). Adversarial attack vulnerability of medical image analysis systems: unexplored factors. Medical Image Analysis, 73(1), 102141.
[20] Yi, J., & Bo, W. (2021). Architecture design of an intelligent monitoring system for turbine filtration device. Journal of Physics: Conference Series, 1944(1), 012040 (6pp).

Edited by: Zhigao Zheng
Special issue on: Graph Powered Big Aerospace Data Processing
Received: Dec 12, 2023
Accepted: Dec 29, 2023