


KGBIAC: Knowledge Graph Based Intelligent Alert Correlation Framework

Wei Wang^(✉) , Rong Jiang, Yan Jia, Aiping Li, and Yi Chen

School of Computer, National University of Defense Technology,
Changsha 410073, China

{wangweil5a, jiangrong, chenyl5a}@nudt.edu.cn,
jiayanjy@vip.sina.com, 13017395458@163.com

Abstract. Alert Correlation is a key part of intrusion detection technique. Traditional methods based on the situation awareness techniques usually store the different dimensions of security information in separate knowledge bases, which leads to the lack of synergies between the various dimensions. For complex attacks, it is difficult to integrate all context information quickly to launch real-time and accurate analysis. To address these issues, we proposed an integrated intelligent security event correlation analysis system, named KGBIAC, which uses knowledge graph to represent and store the network security information. We explain the structure of KGBIAC and conduct an experiment on the DARPA 2000 dataset. Performance evaluation shows that the KGBIAC performs potentially effective.

Keywords: Alert correlation · Knowledge graph · Vulnerability · Cyber security situation awareness

1 Introduction

With the rapid development of computer technology, network viruses, Dos/DDOs and other cyber-attacks are also growing. In order to deal with the increasingly complex and hidden network security threats, it is necessary to integrate the heterogeneous information generated by multi-source security devices with the technology of network security situation to aware the whole network environment. Security event correlation technology provides a solution for the problems above, which integrates the isolated low-level network security event information, and through particular methods to explore the real contact between events [1]. The alert correlation process mainly consists filtering, aggregation and attack scene reconstruction [2]. Security event correlation analysis in traditional Cyber Security Situation Awareness (CSSA) takes into account multiple dimensions of information, such as network infrastructure dimension, vulnerability dimension, and cyber threat dimension [3]. However, there are a number of problems with such systems. First of all, the traditional CSSA systems store the different dimensions of security information in separate relational database, the coordination between the various dimensions of poor ability to launch real-time and accurate analysis. Second, relational database storage is not efficient enough for joint search of multiple dimension information. Third, the traditional rule-based association

analysis needs to rely on expert knowledge to construct the attack scene which lack of ability of reasoning automatically.

To this purpose, in this paper, we present KGBIAC that constructs a network security knowledge graph to fuse independent data into higher-level knowledge. Our framework mainly includes two parts, knowledge graph construction and the use of knowledge graph for correlation analysis. Initially, we fuse network knowledge from a variety of data sources to build a unified knowledge graph based model, which is composed of vulnerabilities kb, network infrastructure kb, cyber threat kb and alerts kb. We also detail the data sources for each dimension. Furthermore, we explain how to connect these sub knowledge graph together to form an intelligent and useful kb. Finally, we conduct experiment on DARPA 2000 dataset and prove the feasibility of our framework.

The remainder of this paper is organized as follows. In Sect. 2, we briefly review relate works. Then, in Sect. 3 we present our proposed framework in detail and present a case study that illustrates the powerful analytic capabilities in KGBIAC, followed by performance analysis in Sect. 4. Finally, we draw our conclusion in Sect. 5.

2 Related Works

For the correlation process, the input data can only one data source or multiple data sources [4]. Obviously, the cost of getting better results using multiple data sources is increasing the complexity of alert correlation systems due to the heterogeneity different input. Y Zhang et al. introduces a simple data fusion technology which prepares a large number of raw security data [5]. These data obtains a standardized asset data set, threat data set, vulnerability data set and network structure data set. They analyzes the relationship between assets, threats, vulnerabilities and security events. Xin Zhuang et al. propose a system Unified Security Information Management Platform (USIM) [6]. In this system, they focus on alerts fusion to reduce the number of alerts, alerts verifying by applying contextual information such as vulnerability information, and alerts correlation with statically built knowledge bases. There is no unified knowledge base model for the description of these elements. To some extent, it limits the flexibility and expansibility of the alert correlation and other CSSA components.

Recent years, some researchers proposed some novel approaches for CSSA based on ontology model. GAO J provides an ontology-based attack model [7]. They categorize attacks into five dimensions, which include attack impact, attack vector, attack target, vulnerability and defense. Afterwards they build an ontology according to these five dimensions and populate the attack ontology with information from many open source information, like NDV, CVE and etc. Finally they propose an ontology-based framework for security assessment and describe the utilization of ontology in the security assessment. Alireza S et al. propose ONTIDS, an ontology-based alert correlation framework that store four dimensions security knowledge, including alert information, current networks context, vulnerability information and attack information in ontologies [8]. They describe the structure of the designed ontology and the detailed attributes for each sub ontology. ONTIDS use SQWRL to correlate and remove non-relevant alerts. Sumit M et al. propose an ontology which comprises three

fundamental classes: means, consequence and targets [9]. And they apply reasoned logic language to find relevant information. However, the common problem of these method is the definition of security elements and indicators is too abstract. In order to ensure that the ontology has good flexibility and versatility, many researchers can only broadly define the concepts and ontology framework, resulting in the practicality and operability are poor.

3 KGBIAC Framework

Our correlation framework includes four layers, multi-source alert collection layer, alert normalization layer, alert correlation layer and correlation results display layer. The alert collection layer collects multi-source security evidence by deploying different security devices such as NIDS Snort, HIDS OSSEC, firewall, and vulnerability scanning tool NMAP and et al. However, the alerts are usually heterogeneous, cannot be used directly. Alert normalization layer converts them to a unified format, usually IDMEF format [10]. Alert correlation layer is the core component of KGBIAC, which includes alert fusion, alert verification and attack thread correlation analysis. Correlation results display layer receives the analysis result and uses front-end framework D3.js to show the final result [11].

Here we will start from the following four aspects. First, we illustrate the knowledge graph tool and query language of our system. Then we introduced in detail the significance of the sub knowledge bases and the source knowledge of them. Furthermore, we integrate the sub knowledge bases as a unified knowledge map. Finally, we describe how to correlate security events based on our knowledge graph.

3.1 Knowledge Graph Tool and RDF Query Language

Knowledge Graph is a knowledge base used by Google to enhance its search engine's search results with semantic-search information gathered from a wide variety of sources. The current Knowledge Graph has been used to refer to various large-scale knowledge bases. Triples is a general representation of a KG. $G = (E, R, S)$ represents whole KG and $E = e_1, e_2, \dots, e_n$ is the collection of entities in the KG, which includes n types entities. $R = r_1, r_2, \dots, r_m$ is the collection of relations in the KG, which includes $|R|$ types relations. $S \subseteq E \times R \times E$ represents the collection of triples in the KG. There is an edge connection between entities and each entity has a set of attributes. Common open KG such as Freebase [12], Wikidata [13] and DBpedia [14].

We employ Blazegraph for our knowledge base tool. Blaze-graph is a high-performance graph database which support for RDF/SPARQL APIs. It supports large scale edges on single server and SPARQL as knowledge query tool. SPARQL (SPARQL Protocol and RDF Query Language) is an RDF query language which has the ability to retrieve and manipulate data stored in Resource Description Framework (RDF) format [15].

3.2 Sub Knowledge Bases

Network infrastructure Knowledge Base collects and stores the overall configuration information of current system environment, including static information and dynamic information. Static information mainly refers to the information that does not change frequently, mainly contains hardware and software. The Official Common Platform Enumeration (CPE) collects the currently known software and hardware specifications and uniquely identifies it through a unified resource descriptor, which is a useful tool to represent the particular operating system or application software. Besides dynamic network infrastructure includes IP address, mac address in the current network environment also need to be saved in KG.

Vulnerability Knowledge Base organizes all the vulnerabilities that have been announced through the form of knowledge graph. Vulnerability is often an important basis for attackers to launch attacks. First, we get all the known vulnerabilities from the National Vulnerability Database (NVD), which contains a lot of CVE items. In addition, we also put Common Vulnerability Scoring System (CVSS) along with the CVE items to KG. CVSS gives a risk score for all vulnerabilities, thus determining the severity of the vulnerability. Almost every CVE entry points to a Common Weakness Enumeration Specification (CWE) entry, which indicates single vulnerability type.

Cyber Threat Knowledge Base enumerates known attack patterns which usually used for exploiting vulnerabilities within the network infrastructure by attackers. CAPEC (Common Attack Pattern Enumeration and Classification) is an important source of knowledge for attack threats. Each CAPEC item describe the attack mode, attack steps, attack threat level and attack response measures. Figure 1 shows the CAPEC Knowledge Graph. Every node represents each attack pattern and the edge indicates the parent-child relationships between the attack patterns.

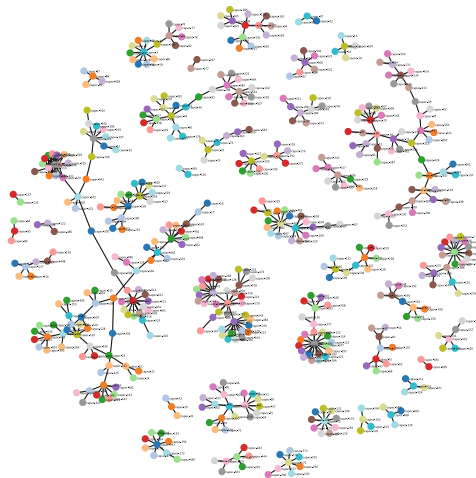


Fig. 1. CAPEC knowledge graph

Alert Knowledge Base. Alert sensors generate alerts based on the abnormal behavior by IDS. Normally used IDS sensors include NIDS and HIDS. NIDS is used more frequently than HIDS. Common NIDS includes snort, Bro, AIDE and etc. We only add Snort alert rules to the Knowledge Graph by now. In future we will fuse more knowledge of the other IDS rules.

3.3 Knowledge Graph Fusion

We have described all of the sub Knowledge Bases in this system, including the Network Infrastructure Knowledge Base, the Vulnerability Knowledge Base, the Cyber Threat Knowledge Base and the Alert Knowledge Base. Next we will illustrate how we can integrate these independent Knowledge Bases into a unified Cyber Security Knowledge Graph.

Each vulnerability refers to a set of CPE items, so there are edges between these two KBs' nodes. In addition, through the vulnerability scanning tools can scan the existence vulnerabilities of the hosts. So we can also connect Vulnerability KB with dynamic Network Infrastructure KB.

Each vulnerability has a reference to CWE and each CAPEC is associated with a number of CWE entries. So we can establish the connection between CAPEC and CVE through CWE, and there is a one-to-many relationship between CAPEC and CVE, because there are many vulnerabilities related to the same type of attack.

Currently, we only store the Snort alert rules, part of them have a clearly reference to particular CVE items. So we can create indirect relationships in the two bases. In the future, we will find others ways to extend the associations between these two KBs.

Figure 2 shows a part of our Knowledge Graph of the system, which depicts the association of these KBs and the attributes of each KB. On the basis of the Knowledge Graph, we can carry out a lot of work on Cyber Security Situation Awareness, such as situation assessment, correlation analysis and etc.

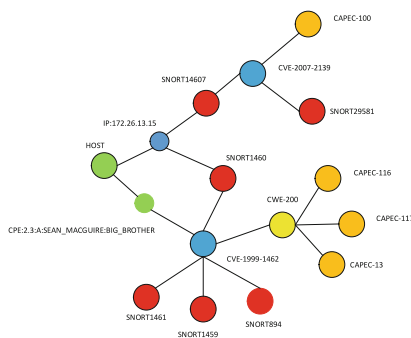


Fig. 2. System knowledge graph framework

3.4 Knowledge Graph Based Event Correlation

Alert Normalization is an important task for tidying the alerts into a unified format and extracting required information. In response to this requirement, the system uses regular expressions for log normalization and information extraction. And then call the detector's information extraction engine. Use the engine to extract important information, such as alert generation time, alert source IP, destination IP, and log description.

Alert Fusion. The main purpose of this phase is to combine alert logs from different detectors but for the same event. The principle of fusing two or more alerts is that if the alerts are generated within a time window, and the attributes of the alerts are consistent. These alert attributes include source IP, destination IP and so on.

Alert verification. The purpose of this step is to filter those unrelated alerts generated by unsuccessful attack. At this point we can use the SPARQL statement, combined with the established Knowledge Graph to query useful knowledge quickly. Assuming the host at this time has a Snort alert, numbered 14607. First through the SPARQL to retrieving all the vulnerabilities of the target host, and then retrieving the vulnerability related to the alert. If the vulnerability in the host vulnerability set, then the alert is the correct alert. Otherwise the alert will be filtered as false alert.

Attack Thread Correlation Analysis. The process of attack thread correlation analysis is based on the existing alerts to predict the real purpose of attackers. Assuming there is an alert Snort-14607, we can intelligently analyze the CVE items associated with the alert as well as other related snort alerts and CAPEC information. Through the Fig. 2 we can find Snort-14607 has a direct reference to CVE-2007-2139 and has an indirect relationship with CAPEC-100 and Snort-29581.

4 Performance Evaluation

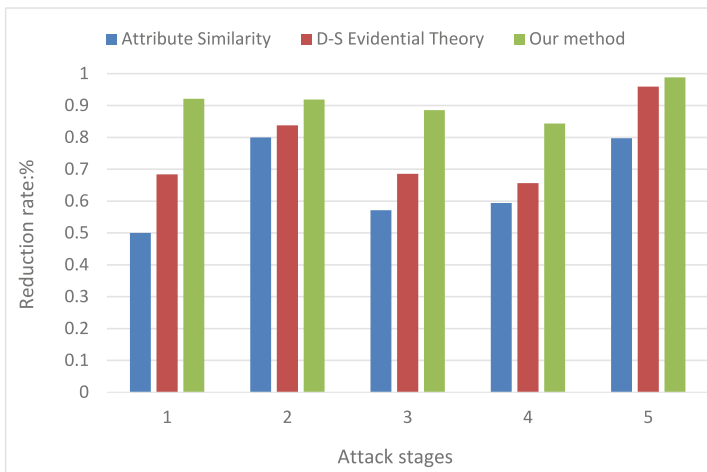
We use known DARPA 2000 as experiment data set and replay LLDDOS 1.0 attack scene. MIT official gives the number of alarms they have collected. In addition, we use Snort to sniff the attack scene. Table 1 lists the number of this two parts alerts. We test the performance of our framework on this dataset and compare with Attribute Similarity Method and D-S Evidential Theory method. These two methods are classical alert correlation methods. Table 2 shows the number of remaining alerts after analysis by three methods. Figure 3 shows the effectiveness of these methods. From the figure we can clearly find our framework achieves the best. Also during the correlation process, Snort sensor detects Snort-1918 which means the event is generated when a scan is detected and Snort-1957 which means an attacker attempts to ping the Remote Procedure Call (RPC) sadmind. Through KGBIAC we found these alerts are related to CVE-1999-0977. This CVE indicates buffer overflow in Solaris sadmind allows remote attackers to gain root privileges. At this point we can infer the attacker's real purpose is through buffer overflow in Solaris sadmind to gain root privileges.

Table 1. MIT alerts number and Snort alerts number

Stage	Attack description	MIT alerts	Reduction ratio
1	Host detection	31	38
2	Vulnerability scanning	32	160
3	System intrusion	35	70
4	Trojan installation	22	32
5	DDos launching	1754	3201

Table 2. Remaining alerts after reduction

Stage	Attribute-Similarity method	D-S Evidential Theory	Our method
1	19	12	3
2	32	26	13
3	30	22	8
4	13	11	5
5	650	130	37

**Fig. 3.** Alert reduction rate

5 Conclusion

In this paper, we proposed an alert correlation framework based on Knowledge Graph. We mainly introduce how to build a cybersecurity knowledge graph and how to use the KG to carry out alert correlation analysis.

In order to make up for the problem of the different dimensions of security information stored in separate knowledge bases, which leads to the lack of synergies between the various dimension, we proposed our KGBIAC framework. Our proposed framework is generic, easy to be adapted by other systems. And can be very flexible to expand the knowledge base. But we only integrate the open source structured cyber

security knowledge by now. In future, we will dig more security related knowledge to expand our Knowledge Graph. Such as by extracting the attack scene from the natural description information. In addition, we will further optimize the alert correlation method to improve accuracy, precision and recall.

Acknowledgements. This work is supported by the National Key Research and Development Program No. 2016YFB0800804, No. 2016YFB0800803, No. 2016YFB0800802

References

1. Liao, H., Lin, C., Lin, Y.: Intrusion detection system: a comprehensive review. *J. Network Comput. Appl.* **36**(1), 16–24 (2013)
2. Valeur, F., Vigna, G., Kruegel, C., Kemmerer, R.A.: Comprehensive approach to intrusion detection alert correlation. *IEEE Trans. Dependable Secure Comput.* **1**(3), 146–169 (2004)
3. Stanton, N.A., Stewart, R., Harris, D., Houghton, R.J., Baber, C., McMaster, R., Salmon, P., Hoyle, G., Walker, G., Young, M.S., et al.: Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. *Ergonomics* **49**(12–13), 1288–1311 (2006)
4. Elshoush, H.T., Osman, I.M.: Alert correlation in collaborative intelligent intrusion detection systems—a survey. *Appl. Soft Comput.* **11**(7), 4349–4365 (2011)
5. Zhang, Y., Tan, X.-B., Cui, X.-L., Xi, H.-S.: Network security situation awareness approach based on Markov game model. *J. Software* **22**(3), 495–508 (2011)
6. Zhuang, X., Xiao, D., Liu, X., Zhang, Y.: Applying data fusion in collaborative alerts correlation. In: *International Symposium on Computer Science and Computational Technology, ISCSCT 2008*, vol. 2, pp. 124–127. IEEE (2008)
7. Gao, J.-B., Zhang, B.-W., Chen, X.-H., Luo, Z.: Ontology-based model of network and computer attacks for security assessment. *J. Shanghai Jiaotong Univ. (Science)* **18**(5), 554–562 (2013)
8. Sadighian, A., Fernandez, J.M., Lemay, A., Zargar, S.T.: ONTIDS: a highly flexible context-aware and ontology-based alert correlation framework. In: Danger, J.-L., Debbabi, M., Marion, J.-Y., Garcia-Alfaro, J., Zincir Heywood, N. (eds.) *FPS-2013. LNCS*, vol. 8352, pp. 161–177. Springer, Cham (2014). doi:[10.1007/978-3-319-05302-8_10](https://doi.org/10.1007/978-3-319-05302-8_10)
9. More, S., Matthews, M., Joshi, A., Finin, T.: A knowledge-based approach to intrusion detection modeling. In: *2012 IEEE Symposium on Security and Privacy Workshops (SPW)*, pp. 75–81. IEEE (2012)
10. Carey, N., Clark, A., Mohay, G.: IDS interoperability and correlation using IDMEF and commodity systems. In: Deng, R., Bao, F., Zhou, J., Qing, S. (eds.) *ICICS 2002. LNCS*, vol. 2513, pp. 252–264. Springer, Heidelberg (2002). doi:[10.1007/3-540-36159-6_22](https://doi.org/10.1007/3-540-36159-6_22)
11. Zhu, N.Q.: *Data Visualization with D3.js Cookbook*. Packt Publishing Ltd., Birmingham (2013)
12. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 1247–1250. ACM (2008)
13. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. *Commun. ACM* **57**(10), 78–85 (2014)
14. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: a nucleus for a web of open data. *The semantic web*, pp. 722–735 (2007)
15. Prud, E., Seaborne, A., et al.: *SPARQL query language for RDF* (2006)