Network Security Situation Awareness Based on Spatio-temporal Correlation of Alarms

Zehua Ren[1], Yang Liu[1]*, Huixiang Liu[1], Baoxiang Jiang[1], Xiangzhen Yao[2], Lin Li[2], Haiwen Yang[3] and Ting Liu[1]
[1] *MoE KLINNS, Xi'an Jiaotong University*, Xi'an, Shaanxi, China
[2] *Cyber Security Research Center, China Electronics Standardization Institute*, Beijing, China
[3] *State Grid Shaanxi Electric Power Company Limited*, Xi'an, Shaanxi, China
* Corresponding author. Email: yangliu@xjtu.edu.cn.

***Abstract*—Traditional intrusion detection systems often deal with massive alarms based on specific filtering rules, which is complex and inexplicable. In this demo, we developed a network security situation awareness (NSSA) system based on the spatio-temporal correlation of alarms. It can monitor the security situation from the temporal dimension and discover abnormal events based on the time series of alarms. Also, it can analyze alarms from the spatial dimension on the heterogeneous alarm graph and handle alarms in batches of events. With this system, system operators can filter most irrelevant alarms quickly and efficiently. The rich visualization of alarm data could also help find hidden high-risk attack behaviors.**

***Index Terms*—Situation Awareness, Spatio-temporal Correlation, Community Discovery, Subgraph Mining, Pattern Matching.**

I. Introduction

Intrusion detection systems (IDSs) have been widely deployed to identify malicious network traffic based on predefined signatures in recent years. However, an IDS would generate huge amounts of false alarms, which prevents system operators from comprehending the network security situation. In the real case, these false alarms are usually eliminated by specific filtering rules, which are manually designed according to the characteristics of each false alarm. This approach is time-consuming and unsuitable for large-scale networks with millions of heterogeneous alarms. Also, it is difficult to migrate these specific rules to other networks.

In this demo, we design a network security situation awareness (NSSA) system based on the spatio-temporal correlation of alarms. First, we use the Exponential Weighted Moving-Average (EWMA [1]) model to locate abnormal events via monitoring multiple temporal statistical indicators of alarms in real-time. Then, we construct alarm graphs within each hour, and deal with alarms in batches based on the spatial correlation of alarms on the graph. With the NSSA system, system operators could handle alarms from IDS in an explainable and efficient approach semi-automatically.

II. System Architecture

As shown in Fig. 1, the NSSA system mainly consists of three parts: data preprocessing, pattern mining, and similarity analysis. The details of each part are as follows.

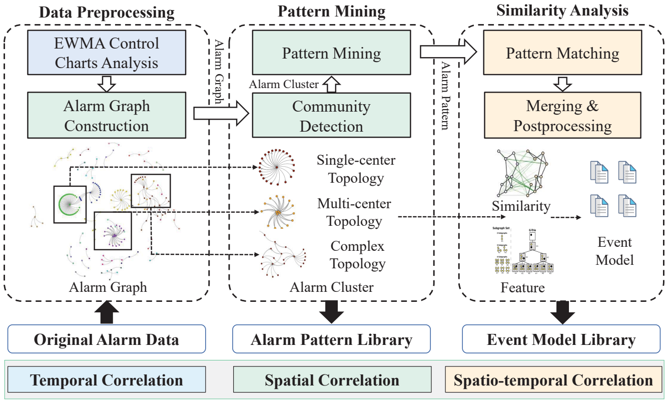


Fig. 1. Overall framework of the NSSA system.

A. Data Preprocessing

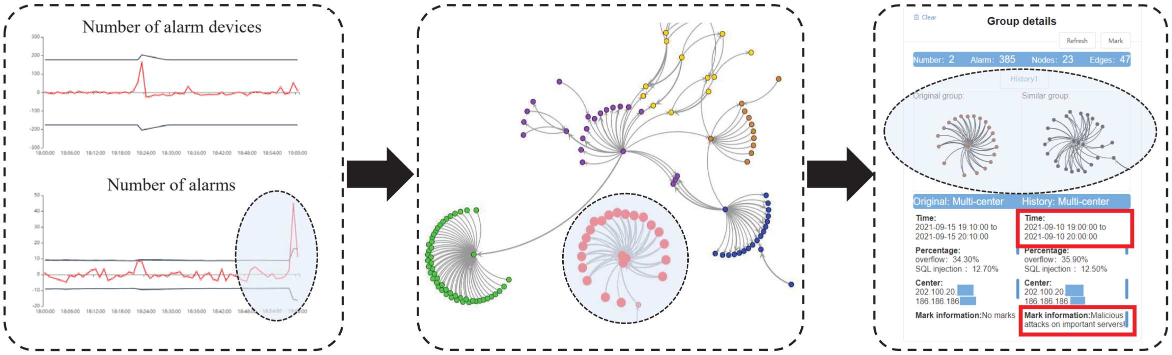
The alarms triggered by the same event would be closely distributed in both temporal and spatial dimensions. First, we use the EWMA control charts to quickly capture the occurred moment of abnormal traffic or access behaviors.

As one event may trigger a series of alarms, it would be cost-effective to aggregate the alarms related to the same event into a group, and handle the alarms in batches of events. Moreover, it could further ease the burden of alarm analysis if we can recommend the historical decision of similar events to system operators. With this idea, we split the alarms by time and construct an alarm graph for each slice, where each node denotes an IP address, and each directed edge denotes an alarm from the source IP address to the destination IP address. The time window is dynamically set according to the current security situation.

The alarm graphs could be filtered by vendors of alarm devices and rendered with different colors based on clusters, vendors, number of alarms, and customized threat indicators in the NSSA system. With the alarm graph, system operators could master the overall network security situation in a visual and friendly way.

B. Pattern Mining

We can find alarm patterns in alarm graphs using improved community discovery algorithms. First, alarms in the graph are roughly clustered into different clusters via the Louvain



Group details

Number: 2 Alarm: 385 Nodes: 23 Edges: 47

Original: Multi-center | History: Multi-center

Time:
2021-09-15 19:10:00 to
2021-09-15 20:10:00

Time:
2021-09-10 19:00:00 to
2021-09-10 20:00:00

Percentage:
overflow: 34.30%
SQL injection : 12.70%

Percentage:
overflow: 35.90%
SQL injection : 12.50%

Center:
202.100.20.[redacted]
186.186.186[redacted]

Center:
202.100.20.[redacted]
186.186.186.[redacted]

Mark information: No marks

Mark information: Malicious attacks on important servers!

(1) Anomaly detection via EWMA control charts.

(2) Event Localization by analyzing alarm patterns in the alarm graph.

(3) Classification based on historical alarm patterns.

Fig. 2. Demonstration of intrusion detection, localization, and automatic classification.

[2] algorithm based on community modularity. Then, the clustering results are amended to obtain the final alarm pattern. Specifically, alarms with rare attack types are picked out and handled separately. Clusters with complex topologies are further split into simple clusters (i.e., single-centered, multi-centered, and other simple topologies) by eliminating a few infrequent connections. Finally, frequent clusters are picked out as alarm patterns and stored in the alarm pattern library.

C. Similarity Analysis

We group alarm patterns with high similarities together into the same event model. The main properties for similarity matching are as follows.

1) *Subgraph motifs* [3]: Topological attributes extracted from alarm patterns by the G-tries algorithm [4].
2) *Alarm types*: Proportion vectors of different attack types for each alarm pattern.
3) *Node homogeneity*: The similarity of central nodes' attributes and the overlap ratio of surrounding nodes for two alarm patterns.

With event models, alarms could be handled conveniently and efficiently in batches. The system also provides a calibration function to revise the automatically generated event models, and the input from system operators will be saved as expert knowledge. After labeling some event models based on specialist knowledge, system operators could handle new alarm patterns in an explainable way semi-automatically.

III. Demo Setup

We have developed an NSSA system based on the above framework. The detailed operations are illustrated in a demo video at **https://bit.ly/NSSA-ST**, where the alarm data are from the IDS of a power grid company. As shown in Fig. 2, after a consecutive attack is launched on three important servers in the network, the NSSA system can detect, locate and classify abnormal events efficiently. **First**, the system presents the overall security situation in the time dimension. Specifically, it monitors the number of alarms and alarm devices in EWMA control charts. Once an indicator crosses the detection boundary, the system will generate an alert indicating some possible abnormal events. As shown in Fig. 2-1, the window of EWMA control charts pops up and alerts, indicating that the number of the alarm exceeds the limit. **Next**, the system can help locate the event in the spatial dimension, i.e., find the corresponding alarm pattern on the alarm graph. As shown in Fig. 2-2, a three-centered alarm pattern can be found in the alarm graph at this moment. **Finally**, the system will automatically match similar historical patterns and sort them according to their similarity. As shown in Fig. 2-3, the most similar pattern in history are displayed in the right column of the "group details". From the figure, these two alarm patterns are extremely similar in both topology structure and alarm type proportion. From the historical data, the previous alarm pattern is automatically labeled as "malicious attacks on important servers". The result is manually verified, and the processing information is "banned attacking IP". Then, all the alarms in the new alarm pattern could be handled in the same way by a simple click in the system.

In addition, for common false positives and low-threat patterns, the system will automatically process them and generate report logs silently. For undecidable patterns, the system will push them to system operators for further investigation. In this way, we can greatly reduce the workload of system operators.

IV. Result Analysis

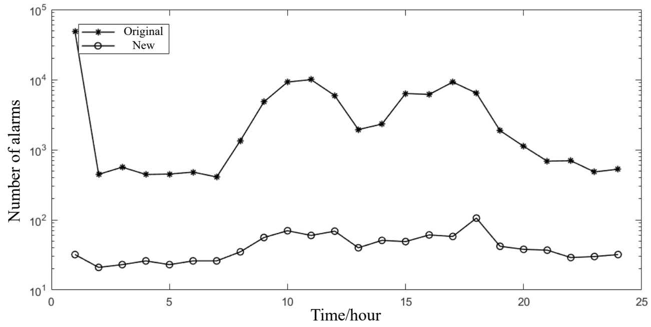


Fig. 3. The effect of reducing the number of alarms.

As shown in Fig. 3, our system can reduce the number of original alarms by 1-3 orders of magnitude. This figure shows the number of alarms received by the system per hour on a given day. Even though the number of alarms at 0:00 is nearly 100,000, we have reduced it to less than 100 based on the similarity and correlation between those alarms. The advantages of our method are fully demonstrated when a large number of alarms occur in a short period (such as DDoS attacks). At this time, system operators can have a good grasp of the overall security situation without being disturbed by a large number of low-risk alarms.

TABLE I
COMPARISON OF OUR SYSTEM WITH ORIGINAL SYSTEM.

Performance	Original system	Our system
Processing time	More than 10”	Less than 1”
Data scale	$10^3 - 10^5/hour$	$10^1 - 10^2/hour$
Accuracy	70%	95%
Cross-platform	No	Yes
Similar matching	No	Yes

Our system has better performance and richer functions than the original system. As shown in Table. I, it shortens the average running time from more than 10 seconds to less than 1 second, greatly improving the real-time performance of the analysis. The number of security events processed by the system is around 100 per hour, and even an alarm flood will not impact the final results. At the same time, our method successfully improves the detection accuracy from 70% to more than 95%. The system can also collect alarms across platforms. It aggregates alarms from different security devices and makes the results more credible. Original systems often use manually set rules to filter out irrelevant alarms. Our greatest contribution is to provide a similar matching function for security events and make the system automatically generate event patterns. It greatly improves the reliability and interpretability of alarm processing and expands the scope of automatic processing by machines.

V. CONCLUSION

We developed an NSSA system with rich data visualization based on spatio-temporal correlation of alarms. It can discover abnormal events using EWMA control charts, and handle alarms in batches of abnormal events based on the alarm graph semi-automatically. Moreover, it can automatically deal with false and low-risk alarms based on historical operations. These functions could help system operators filter useless alarms and focus on hidden high-risk attack behaviors.

REFERENCES

[1] P. Čisar and S. M. Čisar, “EWMA statistics and fuzzy logic in function of network anomaly detection,” *Facta Universitatis, Series: Electronics and Energetics*, vol. 32, no. 2, pp. 249–265, 2019.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[3] S. Haas, F. Wilkens, and M. Fischer, “Efficient attack correlation and identification of attack scenarios based on network-motifs,” in *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2019, pp. 1–11.

[4] P. Ribeiro and F. Silva, “Querying subgraph sets with g-tries,” in *Proceedings of the 2nd ACM SIGMOD Workshop on Databases and Social Networks*, 2012, pp. 25–30.