A multi-step attack scenario mining method based on alert correlation

Jianyi Liu*, Wei Hu², Chan Wang², Jingwen Zhang² and Yahao Zhang²

1School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
2State Grid Information & Telecommunication Branch, Beijing, China
*Corresponding author’s e-mail: liujy@bupt.edu.cn

Abstract: Aiming at the problems of redundant alerts, a multi-step attack scenario mining method is designed based on alert correlation, which has three steps: alerts preprocessing, clustering based on attribute similarity and clustering the attack graph set to mine attack scenario. This algorithm can discover new attack model without a large amount of graph features by analyzing and studying the characteristics and intrinsic relationship of the alerts. Therefore, the full attack scenario can be fully displayed, the efficiency of security managers can be improved.

1. Introduction
With the rapid development of the Internet and the increasingly close network globalization, the network security problem has become increasingly prominent. IDS is deployed in different places in the network, which can complement and cooperate with each other to monitor the security status of the network comprehensively. But the traditional IDS only alarm the single attack [1], and cannot show the whole multi-step attack scenario. Researchers have put forward some methods in the field of complex multi-step scenarios, the most common methods are association based on alarm similarity, predefined attack scenarios, causality, and data mining.

The alarm attribute similarity is the most common and practical method, which does not need prior knowledge, so it is more concise and efficient. Al-saedi [2] used an algorithm based on information gain ratio to extract the similarity of alarm, proposed the CRAF framework to reduce the alarm redundancy and false alarm, and determined the weight of different attributes by information gain ratio. Mohamed [3] et al. proposed an approach to extract three attributes of an alarm by using MD5, Target IO, Signature ID, and timestamp.

Lee [4-6] put forward a method based on Causality Index, which defines the Granger Causality Index between attack steps to judge the Causality between steps, but it needs a lot of parameters to calculate the causality index and to calculate and compare the time series of the alarm. Lin et al. [7] also proposed a real-time intrusion early warning association model based on premises and outcome (Riac). In this model, distributed agents are used to collect early warning information on line, and a premise-result correlation method is used to analyze the attack scenario and intention intrusion. The causal association method is applied to real-time detection. Zhu [8] et al. used a Multilayer perceptron and Support vector machine to estimate alert correlation probabilities by storing the correlation strengths of any two types of alerts in the alert correlation matrix. For the association method based on data mining, its advantage is that it does not need knowledge base and prior knowledge support, but it needs a large amount of computation in the process of data mining, and the accuracy of mining is not as high as that of direct
pattern matching. However, based on the large amount of data today, it is still a promising direction worth studying.

2. Proposed scheme
The multi-step attack scenario mining scheme has three steps: alerts preprocessing, clustering based on attribute similarity and clustering the attack graph set to mine attack scenario. As shown in Figure 1.

2.1. Alarm log preprocessing
The alarm logs generated by different IDS are different, this paper defines the format of alarm log as 11 tuples that each alarm log has, which is (id, time, srcIp, srcPort, dstIp, dstPort, type, priority, gId, groupId, fatherId).

2.2. Alarm clustering based on attribute similarity
After the de-redundancy and normalization of the alarm format, the first association is needed in this part, and the alarm in the same attack process is aggregated together and the related attack graph is used as the source of the mining.

2.2.1. Attribute similarity function
This paper has defined the time dimension, the IP dimension, and the port dimension, carries on the analysis from these three dimensions.

(1) Time dimension: This paper uses the sigmoid function here to calculate the difference in the time dimension between two alerts.

\[ f_{time} = \frac{1}{1 + e^{t}} \quad t = \frac{|t_i - t_j|}{T_p} \]

(2) IP dimension: The ratio of the longest identical prefix IP bits to the total IP bits is used as the IP similarity.

\[ f_{ip} = \frac{n}{32} \]

(3) Port dimension: When the two alarm ports are identical, the similarity of the two alarm ports is set to 1, and the difference is 0.

\[ f_{port} = \begin{cases} 0, & a_i, port! = a_j, port \\ 1, & a_i, port = a_j, port \end{cases} \]

Eventually, the similarity calculation between the two alarms ai and aj is a weighted average of the three dimensions, where \( w_i \) represents the weights of the three features and \( f_i \) represents \( f_{time}, f_{ip} \) and \( f_{port} \), respectively

2.2.2. Attack graph generation
The specific process of the entire process is as follows. Where ASet represents the collection of alarm logs to be clustered, and AGS (Attack Graph Set) represents the collection of ultimately generated candidate attack graphs.

**Input:** The alarm log collection ASet = \{a1, a2, ..., an\}, initializes the AGS = \{\}

**Output:** AGS = \{g1, g2, ..., gn\}, where each graph gi is represented as a candidate attack scene diagram.

(1) Initialize AGS = \{g1\}, where g1 contains a vertex a1.

(2) Remove the unanalyzed alarm ai in ASet to calculate the membership of each graph in AGS = \{g1, g2, ..., gn\}. See step (3) for the specific calculation method.
(3) when calculating the membership degree of a_i in graph g, because the graph contains k vertices,
calculate the similarity C(a_i,a_j) of a_i to each alarm vertex in g, where a_i \in g. And the maximum
similarity is recorded as membership of a_i to g.
(4) Scan each graph in AGS successively, calculate the membership of a_i and each graph, and record
the graph g_k with the largest a_i membership. If its membership is greater than the preset threshold
\theta, then add a_i to graph g_k, that is, add a_i to a new vertex and add a vertex a_i to an edge E(a_i,a_j),
which generates the degree of membership. If all membership is less than the threshold, add an
attack graph with only one vertex a_i.
(5) Follow each alarm in the ASet until the analysis is complete and multiple candidate attack graph
are generated.

2.3. Attack graph clustering

2.3.1. Internal properties of attack graphs
This paper compares the distance between attack graphs by four attributes: time span interval, alarm
interval rate, access rate and difference of exception priority.

(1) Time span interval
This paper defines the time span interval as the interval between the first alert and the last alert
occurrence time.

GTimeDuration(g_i) = |a_i.time - a_n.time|

Defines the distance between two attack graphs as the absolute value of the difference between them.

d_3(g_i, g_j) = |GTimeDuration(g_i) - GTimeDuration(g_j)|

(2) Alarm interval rate
In this paper, the alarm interval rate is defined as the average of the time interval between two
connected alarm vertices in an attack graph.

GIntervalRate(g_i) = \frac{\sum a_i.time - a_j.time}{n - 1}

The distance defined as the alarm interval rate between the two attack graphs is the absolute value of
the difference between them.

d_2(g_i, g_j) = |GIntervalRate(g_i) - GIntervalRate(g_j)|

(3) Access rate
The access rate is defined as the ratio between the number of alarms from outside to inside and the
number of alarms in the whole attack graph.

GIncomeRate(g_i) = \frac{number of income alerts}{number of all alerts}

The distance between the access rates defined as two attack graphs is the absolute value of the
difference between them.

d_4(g_i, g_j) = |GIncomeRate(g_i) - GIncomeRate(g_j)|

(4) The difference of exception priority
The difference of exception priority is defined as the expectation of the difference of priority
between the two alarms.

GPriority(g_i) = \frac{\sum a_j.priority - a_i.priority}{k}

The distance between the two attack graphs defined as the difference between the exception priorities
is the absolute value of the difference between them.

d_1(g_i, g_j) = |GPriority(g_i) - GPriority(g_j)|

2.3.2. Attack graph clustering process
The process of attacking the graph clustering is as follows:
Input: Attack graph collection AGS = \{g_1, g_2, \ldots, g_n\}
Output: A clustered attack graph collection, Cluster = \{set_1, set_2, set_3 \ldots set_m\}. Each set contains several attack graphs of the same type.

1. Initialization of the Cluster = \{set_1\}, set_1 contains an attack graph \(g_1\).
2. Remove the uncategorized attack graph \(g_i\) in AGS and calculate the membership degree of each set in Cluster = \{set_1, set_2, set_3 \ldots set_m\}, see steps (3).
3. When calculating the membership of \(g_i\) belonging to set \(set_i\), because the \(set_i\) contains \(n\) attack graphs, calculate the 4 characteristic distance \(d_k(g_i, g_j)\) of each attack graph \(g_j\) in \(set_i\), where \(g_j \in set_i\). If the distance is less than the threshold \(\delta_1, \delta_2, \delta_3, \delta_4\), then add \(g_i\) to the \(set_i\) collection.
4. The above clustering is performed on each attack graph in AGS until all attack graphs are analysed and the cluster attack graph collection Cluster = \{set_1, set_2, set_3 \ldots set_m\}.

2.4. Attack scenario mining

The attack scene is still represented in the form of a directed acyclic graph, and the final attack scene is \(G\). Take set below as an example to describe the attack scenario generation as follows:

1. Initialize the \(G\)
2. Remove each graph \(g_i\) in the attack graph collection set successively
3. Scan each vertex in the \(g_i\), and if an alarm vertex exists, \(V_j \in g_i\) and \(V_j \in G\), add the vertex to graph \(G\) as a new vertex.
4. Scan each edge in the \(g_i\), and if there is an edge \(E(V_j, V_k)\), from vertex \(V_j\) to \(V_k\), then add an edge from vertex \(V_j\) to \(V_k\) to graph \(G\).
5. Scan all the attack graphs in set and complete the establishment of graph \(G\).

3. Experiment and analysis

3.1. Experimental data

This paper plays the LLDos1.0 data using intrusion detection system snort. The dataset includes inside and dmz and replay 1400 and 3990 IDS logs. During the experiment, 1,400 inside data were used to conduct the experiments. The following figure is extracted from the log collected after snort replay, scanning the host address 202.77.162.213 from the beginning to using the Sadmind exploit tool to taking the victim host permission with the address 172.16.115.20. The log shows that the attacker had six exploit attempts at the victim host. To test the invasion success, the attack script logs in through the telnet attempt as hacker2 after each two intrusion attempts.

3.2. Analysis of results

First, take redundant steps for the alarm. We believe that the same attribute values except for time, which we consider is repeated. After the alarm de-redundancy, the original 1,400 alarms were reduced to 1,135, eliminating 18.93% of the redundant alarms. As shown in Fig. 2 and Figure 3.

Figure 2 Alarm log before de-redundancy.
In the second step, we cluster the alarms through attribute similarity to generate candidate attack graphs. In this process, we need to determine the weights for the time, IP, and port dimensions during clustering.

\[ C(a_i, a_j) = \sum_{i=0}^{n} w_i \cdot f_i \]

Where \( w_1, w_2, w_3 \) represents the weight parameters of time, IP, and port, respectively. \( \theta \) represents the threshold that can be aggregated to the same attack scene.

### Table 1
| Parameter | \( w_1 \) | \( w_2 \) | \( w_3 \) | \( \theta \) |
|-----------|--------|--------|--------|--------|
| Value     | 0.35   | 0.5    | 0.15   | 0.75   |

After clustering the alarm logs, we get 670 attack graphs. Most of these attack graphs are fragment attack graphs with only 1 or 2 alarms. They are not real attack scenarios. Therefore, here, we remove the attack charts containing less than 3 alarms, and finally have 50 attack charts, which contain the attack charts that may represent the attack scene.

In the third step, we cluster the attack graph by computing the four features of the attack graph. The calculated features of the 50 attack graphs are shown in Table 2, with the time span interval and the alarm interval rate in seconds.

### Table 2
| Internal eigenvalues of DARPA2000 attack graph. |
|-----------------------------------------------|
| GPriority | GTimeDuration | GIntervalRate | GIncomeRate |
| 1         | 0             | 1507          | 62.79167    | 0.52       |
| 2         | 0.15          | 5679          | 258.13636   | 0.39       |
| 3         | 0             | 10812         | 600.66667   | 1          |
| 4         | 0             | 1332          | 83.25       | 0.29       |
| 5         | 0             | 7621          | 448.29412   | 0.67       |
| 6         | 0.13          | 3522          | 234.8       | 0.38       |
| 7         | 0.13          | 3506          | 233.73333   | 0.38       |

These four columns represent the four characteristics of time span interval, alarm interval rate, access rate, abnormal priority difference of the attack graph, where the time span interval and alarm interval rate are all in seconds. Next, we aggregate attack graphs with similar features together by an attack graph feature clustering.

For clustering procedures, four parameters need to be set to determine whether the attack graphs are aggregated together. According to the analysis of the alarm log, 1, 2, 3 in snort, and it is reasonable to set \( \delta_1 \) to 0.1 through analysis. Through the analysis of the alarm log of DDoS generated by snort, the whole process is about less than 3 hours, so we set the threshold of the difference of time span interval to 30 mins. According to the analyzed characteristics, for the difference threshold of the alarm interval, we set to 3 mins and the difference threshold to 0.2 for the access rate. Finally, we set the value of \( \delta_1, \delta_2, \delta_3, \delta_4 \) to 0.1, 30min, 3min, 0.2., respectively These four parameters indicate that the difference between the abnormal priority difference is less than 0.2, the time span interval and the 30min, alarm interval is less than the 3min, access rate is less than 0.2, the attack graphs met by the four conditions are clustered together and the clustered attack graph has a similar attack pattern.

After 50 attack graphs are clustered, and the attack scenes are mined, Graphviz is used to draw the graph. Some of the clustered attack graphs contain only one type of attack, so after filtering, a representative attack scenario is as follows:
The attack scenario graph is the result of multiple attack graphs, so it retains only the attack type information of the node, representing an attack pattern. From the diagram, the excavated attack scene diagram is some incomplete attack fragments, or just some scanning operations, do not constitute a complete attack process, less threatening, such as Figure 6 and 7. However, the results can be seen that the method of this paper can dig out the multi-step attack scene in the alarm log to a certain extent, perhaps some of which are not highly threatening scenes, but because the frequency in the security log is relatively high and representative, it can still give some tips and auxiliary effects to the security personnel. In the mining results, the fifth attack scene diagram (Figure 8) represents the process of attacker invasion in the real LLDOSt1.0. And this attack scene diagram is also excavated from the three candidate attack graphs. These three candidate attack graphs contain the process of successful intruder intrusion of the three hosts. They are 202.77.162.213-> 172.16.11 5.20,202.77.162.213-> 172.16.112.10,202.77.162.213-> 172.16.112.50, consistent with the results of the experimental data analysis section.
4. Conclusion
In this paper, we study the reconstruction of multi-step attack scenarios included in the IDS alarm log. In view of the increasingly complex multi-step attacks in network security and the large number of alarm logs difficult to discover the hidden threats, a multi-step attack scene mining method based on the internal characteristics of the attack graph is proposed, realizing the design of the attack scene and each module, and verifying the effectiveness of the proposed algorithm.

Acknowledgments
The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. The work is supported by Science and Technology Project of the Headquarters of State Grid Corporation of China, “The research and technology for collaborative defense and linkage disposal in network security devices” (5700-202152186A-0-0-00).

References
[1] Ramaki A A , Amini M , Atani R E . RTECA: Real Time Episode Correlation Algorithm for Multi-Step Attack Scenarios Detection[J]. Computers & Security, 2014, 49.
[2] Al-Saedi K H , Ramadass S , Almomani A , et al. Collection Mechanism and Reduction of IDS Alert[J]. International Journal of Computer Applications, 2012, 58(4):40-48.
[3] Mohamed A B , Idris N B , Shanmugum B . An Operational Framework for Alert Correlation using a Novel Clustering Approach[C]// International Conference on Communication Systems & Network Technologies. IEEE Computer Society, 2012.
[4] Qin X, Lee W. Causal discovery-based alert correlation[A]. // In Proceedings of the 21st Annual Computer Security Application Conference[C], 2005: 33-40.
[5] Qin X, Lee W. Statistical causality analysis of INFOSEC alert data[A]. // In Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection[C], 2003: 591-627.
[6] Qin X, Lee W. Discovering novel attack strategies from INFOSEC alerts[A]. // In Proceedings of the 9th European Symposium on Research in Computer Security[C], 2004: 439-456.
[7] Lin Z , Li S , Ma Y . Real-time intrusion alert correlation system based on prerequisites and consequence[C]// International Conference on Wireless Communications Networking & Mobile Computing. IEEE, 2010.
[8] B. Zhu, A.A. Ghorbani, Alert correlation for extracting attack strategies,International Journal of Network Security 3 (2) (2006) 244–258.