Journal of Communications and Information Networks
Vol.1, No.4, Dec. 2016
DOI: 10.11959/j.issn.2096-1081.2016.014

Research paper

A multi-step attack-correlation method with privacy protection

ZHANG Yongtang[1, 2], LUO Xianlu[1], LUO Haibo[1]

1. Department of Computer Science and Technology, Guangdong Neusoft Institute, Foshan 528225, China
2. Jiangxi Microsoft Technology Center, Nanchang 330003, China

Abstract: In the era of global Internet security threats, there is an urgent need for different organizations to cooperate and jointly fight against cyber attacks. We present an algorithm that combines a privacy-preserving technique and a multi-step attack-correlation method to better balance the privacy and availability of alarm data. This algorithm is used to construct multi-step attack scenarios by discovering sequential attack-behavior patterns. It analyzes the time-sequential characteristics of attack behaviors and implements a support-evaluation method. Optimized candidate attack-sequence generation is applied to solve the problem of pre-defined association-rule complexity, as well as expert-knowledge dependency. An enhanced *k*-anonymity method is applied to this algorithm to preserve privacy. Experimental results indicate that the algorithm has better performance and accuracy for multi-step attack correlation than other methods, and reaches a good balance between efficiency and privacy.

Key words: network security; multi-step attack; intrusion detection; sequential pattern; privacy protection; data mining

Citation: ZHANG Y T, LUO X L, LUO H B. A multi-step attack-correlation method with privacy protection[J]. Journal of communications and information networks, 2016, 1(4): 133-142.

1 Introduction

In recent years, global security threats, e.g., the Code Red II worm[1], My Doom[2], and DDoSs (Distributed Denial of Service-attacks), have been growing. This creates an urgent need for mutual cooperation between different organizations to resist these attacks and share security-incident alarm data for security analysis. However, if you don't share the alarm data privacy protection, it is easy to be the attacker, lead to leak sensitive information data owners.

Privacy protection generally refers to the protection of an individual or organization, excluding public information, data, etc. In a specific application, data privacy preserves sensitive information that the owner is not willing to disclose or share with others, including characteristics of the sensitive data and the data representation. Normally, privacy refers to sensitive data, e.g., a company's financial situation, one's personal property status, a patient's medical records, etc.

At present, most of the multi-step attack-correlation methods are carried out based on original police reports; however, when a number of different organizations fail to provide alarm data because of privacy protection, the multi-step attack alarm-

Manuscript received Apr. 20, 2016; accepted Aug. 12, 2016
This work is supported by the Ordinary University Innovation Project of Guangdong Province (Nos. 2014KTSCX212, 2014KQNCX24).

data association results may be affected because of the lack of accurate information. Therefore, effectively associating privacy-protection technology with a multi-step attack method requires a better balance between alarm-data privacy and availability requirements. This significantly affects research, whether in theory or in practice.

Since 2000, research on the multi-step attack correlation analysis of alarm data has gradually increased, and achieved a number of results. Mange F[3] proposed a causal relationship between attack analysis and attack association. In 2014, Ning P[4] proposed using the attack steps between the prerequisites and consequences to construct attack scenarios.

Debar H, et al.[5,6] used similar methods in the Mirador project with Prolog predicate logic to describe an attack, and automatically generated association rules according to the description of the prerequisites and consequences. Through the association rules, they discovered a relationship between the reports used to construct attack scenarios. This type of method uses the causal relationship of the different attack steps to define attack-association rules; however, because the rule definition of report is very complex, the design requirement is very high, more difficult, so for the unknown attack is difficult to find the cause-and-effect relationship, there is a defect.

In 2009, Wu B[7] proposed using the GCT (Garbage Collection Time) statistical timing algorithm to mine attack-scene fragments, and then connect the attack fragments into a complete attack scene. However, this method relies heavily on domain and expert knowledge, and it is difficult to find a complete attack scenario. In 2006, Wang K[8] summarized the advantages and disadvantages of the above two types of multi-step attack-correlation methods. He proposed drawing on a sequential pattern-mining multi-step attack-correlation method. The method first converts the alarm database to a global attack sequence; then, the global attack sequence is classified as a candidate attack sequence set within the attack-scenario time window. Finally, from the concentrated candidate attack sequence, he extracts the largest sequential attack-behavior pattern.

In recent years, the multi-step attack — the correlation research in the field of intrusion detection and prevention research is a hot topic. Ringer H A[9] to study the alarm data, and found the alarm data sharing may encounter attack. He suggested to use a hash function or anonymous hash function of sensitive alarm properties, to a certain extent, the protection of privacy alarm data. However, because of the limited alarm-attribute value, this method can easily be abused.

Ning P[4] summarizing other methods, e.g., Kissncr L[10] proposed using a hierarchy to anonymize sensitive attribute values to protect the privacy of sensitive data. This method introduces the concept of discrete attributes with a continuous attribute-hierarchy generation measure. By modifying the similarity function, it redefines the relationship, based on the attack probability, and applies privacy protection methods to the alarm data after the alarm correlation.

However, this method has some problems, e.g., the original similarity function should be modified for different sensitivity properties. If the original similarity function of a sensitive attribute is more complex, then modifying the attribute-similarity function will become very complicated. Further, redefining the relationship based on the attack probability could introduce correlation errors; thus, increasing the number of false alarms in the attack scene graph.

Privacy-protection technology associated with multi-step attack methods has a long history in its field of study. However, combining the two for a specific application, choosing appropriate privacy-protection methods while effectively avoiding privacy information, and providing good information support

for information sharing, have become important research topics for many scholars and experts.

Regarding the problems mentioned above, a PPMAC (Privacy-Preserving Multi-step Attack Correlation) algorithm is proposed in this paper to discover sequential attack-behavior patterns in a protected alert set. By attempting to determine the sequences for global multi-step attacks, this method can be used to predict invaders' future moves, and thus take proper actions to efficiently decrease the impact of such intrusions. With an affordable computation cost, PPMAC has been proven to be more efficient than traditional methods in attack scenario construction while protecting sensitive information in the alert set.

2 PPMAC briefing

This section describes the PPMAC framework, as illustrated in Fig.1.

Network domains are on the left side. The PPMAC engine, a secure third-party multi-step attack correlation processor, as well as an analyzer, are on the right side, interacting with the network domain through the function units of the privacy-preserving agent and monitor.

The original intrusion alerts detected in the network domains are stored in the intrusion event database, which transfers the raw alerts into the data cleaner. The data cleaner is used to reduce the number and enhance the quality of these alerts. After sanitizing the refined data received from the data cleaner, the privacy-preserving agent submits the data to the PPMAC engine for multi-step attack correlation. Finally, the monitor receives the correlation results of attack scenarios from the PPMAC engine for further security response by the network administrators.

The PPMAC engine is composed of four functional units: (1) The Global Attack-Sequence Generator generates a global sequence of attack behaviors ordered by the start-time attribute of the alerts, which are mapped onto the attack-signature identifiers of

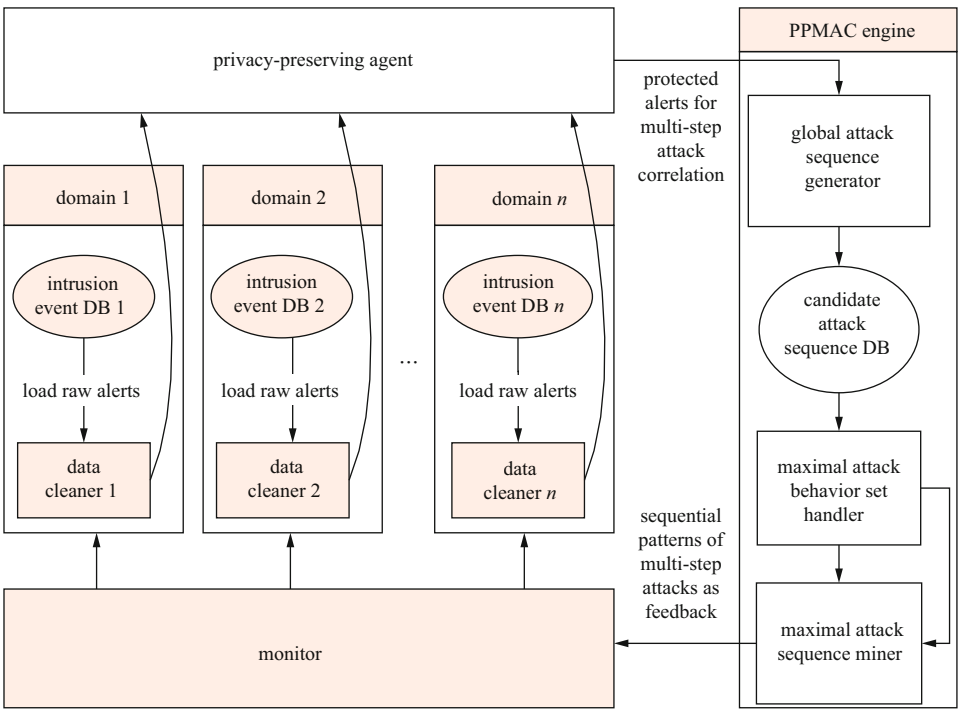


Figure 1 PPMAC framework

consecutive integers. (2) The Candidate Attack-Sequence DB stores the selected candidate attack sequences from the global attack sequence. (3) The Maximal Attack-Behavior-Set Handler produces all the maximal attack behaviors with minimum support, using a pre-defined parameter. (4) The Maximal Attack-Sequence Miner produces multi-step attack scenarios from the protected alert set based on attack-behavior sequence analysis. The mining results can be fed back to the network domains through the monitor to defend against complex intrusions.

Considering the different sensitivity levels of various attributes in the alert, this paper improves the *k*-anonymity method, a classical privacy-preserving method, to preserve the alerts' semantics as much as possible during the process of attribute anonymization.

3 PPMAC algorithms

The method of mining sequential attack-behavior patterns is based on the concept that the various multi-step attack behaviors initiated by attackers always appear in ordered sequences. That is, attackers will logically perform a certain action before carrying out the next move. The LLDoS (Lincoln Laboratory Denial of Service) 1.0 attack scenario[11] explicitly illustrates the general character of multi-step attacks. Consequently, sequential-alert pattern analysis can be used to correlate various suspicious activities that, if considered separately, would not provide sufficient evidence to detect complicated multi-step attacks.

The following terms are defined to clarify the PPMAC Algorithm:

Definition 1 Attribute vector: The attribute vector of each intrusion alert is defined as: {date, start-time, protocol, source-IP, destination-IP, source-port, destination-port, attack-type}, with a start-time vector $\boldsymbol{al} = \{at_1, \cdots, at_8\}$.

Definition 2 Alert set: These are triggered by IDSs (Intrusion-Detection Systems) when intrusion events are detected. Let ***AL*** be the alert set with a finite number of attribute vectors ***al***.

Definition 3 *k*-anonymity: Let $\boldsymbol{sd} = \{sd_1,\cdots,sd_i\}$, $0<i \leqslant 8$ be a vector of selected attributes from ***al***, which are the sensitive items to be protected, $\boldsymbol{sd} \subseteq \boldsymbol{al}$. ***SD*** is the set of ***sd***, as a subset extracted from ***AL***, i.e., $\boldsymbol{SD} \subseteq \boldsymbol{AL}$. ***AL*** satisfies *k*-anonymity if and only if each vector in ***SD*** accordingly appears with at least *k* occurrences.

Following is our PPMAC algorithm.

Step 1 Protect sensitive alert attributes using the improved *k*-anonymity method.

One of the major privacy-preserving methods generally uses a quasi-identifier to support judging whether *k*-anonymity properties are met. When preserving privacy in intrusion alerts, the sensitive attributes are the private items, not the quasi-identifier.

The *k*-anonymity method is applied to selected sensitive attributes in ***SD***, while all other attributes remain unchanged. The discrete values of sensitive alert attributes in ***AL*** are replaced by their generalized values, which are predefined based on the domain's security policy. For example, the original destination-IP, one of the main private attributes, is generalized to its subnet address with a 24-bit mask or even a 16-bit mask. Uncertainty is introduced into the alert set, but the alert regulations are kept for mining.

The generalized alert set $\boldsymbol{AL}_G$, as the output of the privacy-preserving agent, will be sent to the PPMAC engine for multi-step attack-correlation analysis.

Step 2 Multi-step attack-correlation algorithm.

This step discovers maximal attack sequences, based on an Apriori-like theory.

1) Mapping and grouping. To make the mining process more efficient, a one-to-one mapping table between the attack-type strings and a series of consecutive integers is established; i.e., the corresponding integers represent the attack-signature identifier. All values of the attack-

type alerts in $\boldsymbol{AL}_G$ are converted to integers according to the mapping rule.

Definition 4 Attack set $\boldsymbol{S}$ $\{s_1, s_2, \cdots, s_g\}$: The set of attack-signature identifiers according to the attack types reported by the IDSs.

Alert attributes generally have some relationships among the different steps of a multi-step attack process; e.g., the same destination port number or the same network path to the destination IP address. All alerts in $\boldsymbol{AL}_G$ are classified into groups by their protected destination-IP segments, which can recognize single-hop and multi-hop attacks.

2) Construct a global attack sequence. After mapping and grouping, the global attack-sequence generator generates a global attack sequence.

Definition 5 Attack sequence $<a_1, a_2, \cdots, a_n>$: $\boldsymbol{A}$ sequence of attack behaviors, where a_i $(1 \leqslant i \leqslant n) \in \boldsymbol{S}$.

We sort all alerts in each group by ascending start-time after the mapping and grouping procedure to organize the global attack sequences by groups.

3) Produce a candidate attack-sequence set. An attack scenario is a collection of intrusion events that occur closely within a pre-defined time window. Therefore, an iterative process is designed to retrieve the candidate attack-sequence set: a) Initialize the time window to begin at the start-time of the first alert in the global attack sequence. b) Select all alerts in which the start-time is in the scope of the time window. c) Move the time window to the start-time of the next alert. d) Repeat steps a) to c) until the end of the time window reaches the end of the sequence.

After applying the process for each global sequence, the attack steps that fall into the same attack-scenario time window form a candidate attack sequence. The candidate attack-sequence set is obtained by combining the attack sequences, and then is stored in the sequence matrix database.

The set of candidate attack sequences is denoted as *CAS*, and the candidate attack sequences in *CAS* are denoted as cas_p, $1 \leqslant p \leqslant TS$, where *TS* is the total number of candidate attack sequences.

4) Obtain maximal attack-behavior set.

Definition 6 Sequence support: Given two attack sequences $A=<a_1, a_2, \cdots, a_n>$ and $B=<b_1, b_2, \cdots, b_m>$ $(m \geqslant n)$, if $a_1=b_1$ and A is a subsequence of B (A can be obtained by deleting some data from B without changing the order). Sequence A is contained in sequence B, or sequence B supports sequence A, denoted as $B \uparrow A$.

In a set of attack sequences, an attack sequence is maximal if it is not contained in any other sequences. The key target of mining sequential attack behavior patterns is to find the maximal attack sequences among all candidate attack sequences by group. Each of the maximal attack sequences will represent an attack scenario. The support-evaluation method is applied to determine the maximal attack sequences.

Definition 7 Support of attack sequence SUP_{AS}: The ratio between the number of candidate attack sequences supporting attack sequence A, denoted as *CS*, and the total number of candidate attack sequences, denoted as *TS*, is given:

$$SUP_{\mathrm{AS}}(A) = \frac{CS}{TS} \times 100\% . \tag{1}$$

Definition 8 Support of attack behavior SUP_{AB}: The ratio between the number of occurrences of a certain attack behavior a_i in the global attack sequence, denoted as *AB*, and the number of total behaviors contained in this global attack sequence, denoted as *TB*, is given in Eq.(2),

$$SUP_{\mathrm{AB}}(a_i) = \frac{AB}{TB} \times 100\% , \quad (1 \leqslant h \leqslant g) . \tag{2}$$

Attack behaviors that satisfy the predefined minimum support threshold (*min_sup*) are called maximal attack behaviors.

The procedure to obtain the maximal attack behavior set is also the process to produce an attack

sequence with only one behavior (1-sequence), denoted as L_1. The maximal attack-behavior set handler produces L_1, as shown in Algorithm 1. The total number of maximal attack behaviors is denoted as TA.

Algorithm 1 Pseudocode for retrieving L_1

```
for //each attack behavior a_i in attack set S
  If SUP_AB(a_i) ⩾ min_sup
      then a_i belongs to L_1
    end if
end for
```

As described above, we can conclude that the elements in the maximal attack sequences must be the ones in the maximal attack-behavior set.

5) Discover maximal attack sequences. The maximal attack-sequence miner looks for small attack scenarios, and finds progressively larger ones. This process is composed of two procedures: producing a candidate maximal attack-sequence set C_y and finding a corresponding maximal attack-sequence set L_y, where y denotes the length or the number of attack behaviors in the maximal attack sequence. This process ends when no new sequence can be derived from L_y to C_{y+1}. It discovers all the maximal attack sequences of L_y in the given intrusion-event sequences.

a) Assume L_{y-1}, as shown in Eq.(3).

$$L_{y-1}=\begin{cases}\begin{bmatrix}a_{11}\\ a_{21}\\ \vdots\\ a_{x1}\end{bmatrix}, y=2, x\leqslant TA\\ \begin{bmatrix}a_{11} & a_{12} & \cdots & a_{1(y-1)}\\ a_{21} & a_{22} & \cdots & a_{2(y-1)}\\ \vdots & \vdots & \vdots & \vdots\\ a_{x1} & a_{x2} & \cdots & a_{x(y-1)}\end{bmatrix}, y\geqslant 3, x\leqslant TA\end{cases}. \quad (3)$$

The candidate maximal attack-sequence set C_y is generated by joining the maximal attack-sequence patterns in the previous pass, as shown in Eq.4.

$$C_y=\begin{cases}\begin{bmatrix}a_{11} & a_{21}\\ a_{11} & a_{31}\\ \vdots & \vdots\\ a_{11} & a_{x1}\\ a_{21} & a_{31}\\ \vdots & \vdots\\ a_{21} & a_{x1}\\ \vdots & \vdots\\ a_{(x-1)1} & a_{x1}\end{bmatrix}, y=2\\ \begin{bmatrix}a_{11} & a_{12} & \cdots & a_{1(y-2)} & a_{1(y-1)} & a_{2(y-1)}\\ a_{11} & a_{12} & \cdots & a_{1(y-2)} & a_{1(y-1)} & a_{3(y-1)}\\ \vdots & \vdots & & \vdots & \vdots & \vdots\\ a_{11} & a_{12} & \cdots & a_{1(y-2)} & a_{1(y-1)} & a_{x(y-1)}\\ a_{21} & a_{22} & \cdots & a_{2(y-2)} & a_{2(y-1)} & a_{3(y-1)}\\ a_{21} & a_{22} & \cdots & a_{2(y-2)} & a_{2(y-1)} & a_{4(y-1)}\\ \vdots & \vdots & & \vdots & \vdots & \vdots\\ a_{21} & a_{22} & \cdots & a_{2(y-2)} & a_{2(y-1)} & a_{x(y-1)}\\ \vdots & \vdots & & \vdots & \vdots & \vdots\\ a_{(x-1)1} & a_{(x-1)2} & \cdots & a_{(x-1)(y-2)} & a_{(x-1)(y-1)} & a_{x(y-1)}\end{bmatrix}, y\geqslant 3.\end{cases} \quad (4)$$

The number of rows in C_y is denoted as $|C_y|$ and the candidate maximal attack sequences in C_y are denoted as

$$C_y(q),\ 1\leqslant q\leqslant |C_y|. \quad (5)$$

b) Scan the candidate attack-sequence set from top to bottom and read one attack sequence each time. The support count of each candidate maximal attack sequence in C_y increases while it is contained in the attack sequence. We obtain the maximal attack sequences of L_y by deleting those candidate sequences whose support count is less than *min_sup*. The pseudo code for retrieving L_y is shown in Algorithm 2.

Algorithm 2 Pseudo code for retrieving L_y

```
for //each candidate attack sequence CAS(p) in CAS
  for //each candidate maximal attack sequence C_y(q) in C_y
    if CAS(p) ↑ C_y(q)
      then SUP_AS(C_y(q)) increases
    end if
  end for
end for
for //each candidate maximal attack sequence C_y(q) in C_y
  if SUP_AS(C_y(q)) ≥ min_sup
    then C_y(q) belongs to L_y
  end if
end for
```

Meanwhile, if the first element in a candidate attack sequence does not match with any first element in C_y, this candidate attack sequence is demonstrated to not support any sequences in C_y. The sequence is then flagged in the candidate attack sequence set and will not be involved in the rest of the scanning process for producing L_{y+1}. This pruning technique is used to optimize the PPMAC algorithm, aiming to reduce the number of candidate sequences in the subsequent passes.

c) The algorithm ends when no new sequence can be derived from L_y to C_{y+1}.

All the maximal attack sequences discovered by the PPMAC algorithm will be fed back to the network domains through the monitor.

4 Experimental results

This section includes the validation of the effectiveness and scalability of PPMAC for multi-step attack correlation, and then compares its performance with two traditional sequential pattern-mining algorithms: AprioriAll and GSP (Generalized Sequential Patterns). All the experiments are processed with DEF CON 19th (The world's most famous hacker conference) datasets[12], one of the most authoritative attack-scenario datasets from the DEF CON organization.

Experiments were performed in an intranet environment, where TCP Replay, an open-source replaying toolkit, was applied to import the network flows into the deployed Snort network intrusion prevention and detection system[13]. Tab.1 summarizes part of the raw intrusion-alert information.

Table 1 Raw intrusion alerts (partial)

type of intrusion alert	number of alerts before correlation
bad-traffic loopback IP	42
bad-traffic SYN to multicast address	150
bad-traffic TCP port 0 traffic	218
TCP port sweep	5
SNMP request UDP	687
fragmentation overlap	91
SCAN FIN	240
community SIP TCP/IP message flooding directed to SIP proxy	136
ICMP icmpenum	791
bad-traffic loopback traffic	49
SNMP message community string attempt	327
DDOS mstream client to handler	785

4.1 Effectiveness

These experiments were designed to evaluate PPMAC's capability to discover multi-step attack behavior sequences.

We collected the original experimental data from an experimental data set containing 5 290 intrusion alarm records. Tab.2 shows the experimental intrusion-alarm data after the original intrusion alarm and privacy protection. In the experiment, destination-IP and destination-port were considered as the main private attributes, which could show the intranet topology and the services running on it. These

Table 2 Example of raw intrusion alerts and protected intrusion alerts (k = 2)

protocol	source-IP	source-port	saw intrusion alerts		protected intrusion alerts (k = 2)		attack-type
			destination-IP	destination-port	destination-IP	destination-port	
TCP	10.10.1.20	44 355	10.10.2.254	8 080	10.10.*.*	*	community SIP TCP/IP message flooding directed to SIP proxy
TCP	147.4.10.50	166	127.2.19.7	2 844	127.2.19.*	*	bad-traffic SYN to multicast address
TCP	110.121.50.119	25 978	127.2.19.20	2 844	127.2.19.*	*	bad-traffic SYN to multicast address
TCP	123.122.12.120	43 345	10.10.3.254	63 322	10.10.*.*	*	bad-traffic loopback traffic

attributes were transformed by the privacy-preserving agent according to the improved k-anonymity method with k = 2.

PPMAC's multi-step attack-correlation results are shown in Tab.3. We only compared it with the GSP algorithm, as AprioriAll algorithm discovered too many attack-scenario fragments, which would result in exhausting computation.

Table 3 Comparison of attack scenario recognition (k = 2)

comparison items	PPMAC	GSP	AprioriAll
number of real attack scenarios	47	49	71
number of discovered attack scenarios	46	43	52
number of correct attack scenarios	44	38	39
false positive ratio	8.4%	11.6%	25%
false negative ratio	12.1%	20.5%	45.1%

PPMAC achieved comparatively better results for the correct multi-step attack scenarios on the protected datasets; this shows no capability loss for discovering attack scenarios after the privacy preserving process. Because the attack type attributes are fully involved in the process of discovering attack behavior sequences. Meanwhile, PPMAC has a lower false-positive ratio and false negative ratio than GSP, which we attribute to our novel method of producing candidate maximal attack sequences.

4.2 Scalability

In the experimental dataset, we used the range from 1% to 0.25% as the minimum support value (min-sup). The runtimes of PPMAC, GSP, and AprioriAll within the assigned min-sup range are shown in Fig.2.

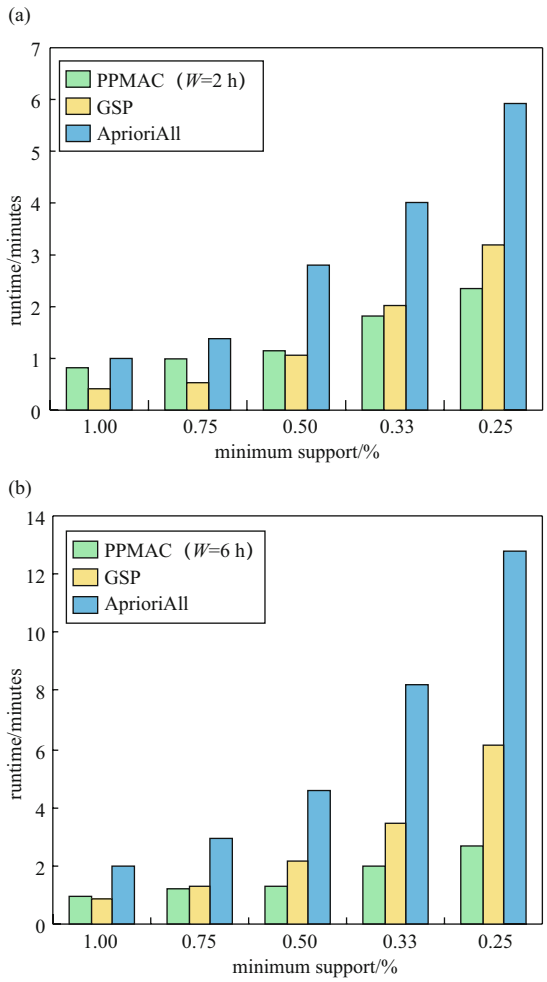


Figure 2 Scalability: min-sup's effect on the runtime with different methods. (a) Attack scenario time window is 2 h. (b) Attack scenario time window is 6 h.

As proven in the experiments, PPMAC averages around 10% to 500% faster than AprioriAll. When W is smaller and *min_sup* is higher, GSP is around 30% faster than PPMAC, as PPMAC must protect the sensitive attributes in the original alerts. With the decreasing *min_sup*, more maximal attack behaviors in the sequence, as well as more candidates maximal attack sequences, are generated. PPMAC's performance thus benefits from its optimized method of producing candidate maximal attack-sequence sets. The runtime gap between the two methods increases. Generally, PPMAC's total runtime increases with a decreasing *min_sup*.

Fig.3 shows PPMAC's execution time for finding maximal attack sequences in the experimental dataset, with various attack-scenario time windows and min-sup values. Generally, increasing the time window leads to extending the length of the candidate attack sequences, while decreasing min-sup leads to the growth of initial maximal-attack behaviors. The joint effort of both factors should be held liable for the rapid increase in processing time.

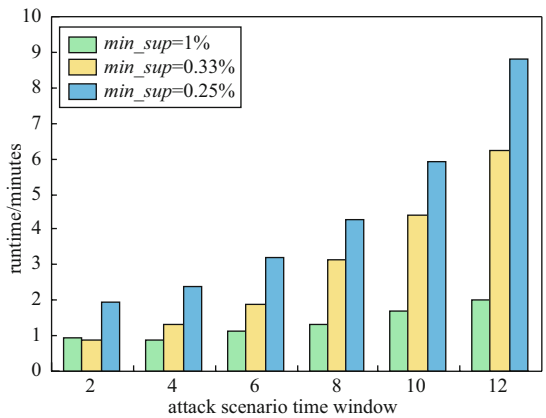


Figure 3 Scalability: time window/support versus runtime in PPMAC

The effectiveness of the number of alerts versus the runtime is shown in Fig.4. It illustrates that the time cost is roughly linear, despite the different ranges of alert numbers.

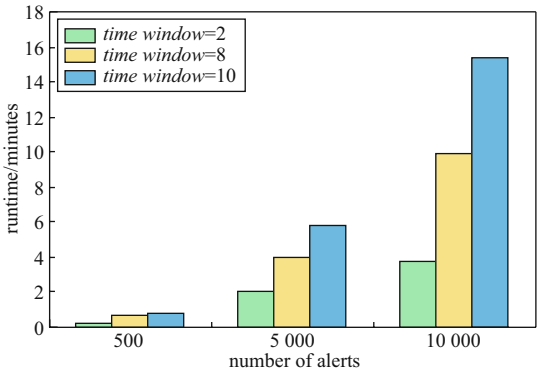


Figure 4 Scalability: PPMAC linear performance in different situations (*minimum support* = 0.25%)

5 Conclusions and future work

Sequential pattern-mining techniques are considered an essential part of the alert correlation and analysis field. In this paper, we proposed a novel method, PPMAC, which could discover multi-step attack behavior patterns while preventing the risk of sensitive-information disclosure during the process. On this basis, experiments were conducting and testing data proved that PPMAC had obviously better performance and accuracy in recognizing multi-step attack scenarios.

Privacy guarantees will accelerate the research developments in the corresponding area, and make it more feasible to discover hidden multi-hop attacks across network domains, which could be a promising direction for future research.

Meanwhile, the attack time window is usually pre-defined according to off-line analysis of the attack data. However, considering the flexible time expense regarding various attack types, whether a scalable time window can be applied to more efficiently produce candidate attack sequences to discover multi-step attack scenarios remains an open question.

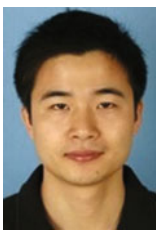
References

[1] ZHANG S, LI J, CHAN X, et al. Building network attack graph for

alert causal correlation[J]. Computers & security, 2008, 27(5-6): 188-196.
[2] ZHOU J, HENCHMAN M, REYNOLDS B, et al. Modeling network intrusion detection alerts for correlation[J]. ACM transactions on information and system security, 2007, 10(1): 4-34.
[3] MANGE F. Multistep attack detection and alert correlation in intrusion detection system[J]. Information security and assurance, 2011, 200: 101-110.
[4] NING P, CUI Y, REEVES D S, et al. Techniques and tools for analyzing intrusion alerts[J]. ACM transactions on information and system security, 2014, 7(2): 273-318.
[5] DEBAR H, WISPY A. Aggregation and correlation or intrusion-detection alerts[J]. Lecture notes in computer science, 2011, 2212: 85-98.
[6] MORIN B, DEBAR H. Correlation of intrusion symptoms: an application of chronicles[J]. Lecture notes in computer science, 2013, 2820: 91-112.
[7] WU B. Analysis of alert correlation based on sequential pattern in intrusion detection[J]. Journal of university of electronic science and technology of China, 2009, 3: 38-42.
[8] WANG K, SOLO S J. Privacy-preserving payload-based correlation for accurate malicious traffic detection[C]//The 2006 SIGCOMM Workshop on Large-scale Attack Defense, Pisa, Italy, 2006: 99-106.
[9] RINGER H A. Privacy-preserving collaborative anomaly detection[D]. Princeton: Princeton University, 2009.
[10] KISSNCR L, SONG D. Privacy-preserving set operations[J]. Lecture notes in computer science, 2005, 3621: 231-257.
[11] SRIKANT R, AGRAWAL R. Mining sequential patterns: Generalizations and performance improvements[J]. Lecture notes in computer science, 2012, 1057: 1-17.
[12] YCGNCSWARAN V, BRADFORD P, JHA S. Global intrusion detection in the domino overlay system[R]. University of Wisconsin-Madison Department of Computer Sciences, 2010.
[13] AUGURAL G, FODOR T. Approximation algorithms for k-anonymity[J]. Journal of privacy technology, 2015, 20(1): 12-23.

About the authors

ZHANG Yongtang [corresponding author] was born in 1981. He earned a Master's degree in network and communications engineering in 2005 from the Central China Normal University. He is currently working in the Guangdong Neusoft Institute as an associate professor, and as a systems analyst at the Jiangxi Microsoft Technology Center. At present, his main research interests are communications and wireless sensor network applications, network security, and offensive and defensive techniques. (Email: gov211@163.com)



LUO Xianlu was born in 1973. He earned a Master's degree in computer applications in 2002 from Northeastern University. He is currently working in the Guangdong Neusoft Institute as an associate professor and systems analyst. At present, his main research interests are algorithms, network security, and offensive and defensive techniques. (Email: luoxl@neusoft.com)

LUO Haibo was born in 1980. In 2009, he graduated with a Masters in computer science and computer application engineering from Wuhan University. He is currently in Guangdong Neusoft Institute, as a lecturer. His research interests are information-security research and data mining. (Email: luo-hb@neusoft.com)