An IDS Alert Fusion Approach Based on Happened Before Relation

Ming XU, Ting WU, JingFan TANG
Institute of Computer Application Technology, College of Computer
Hangzhou Dianzi University
Hangzhou, P.R. China
mxu@hdu.edu.cn

***Abstract*—Alert fusion is a process that analyzes the alerts produced by one or more intrusion detection systems and provides a more succinct and high-level view of security event or attempted intrusions. Unfortunately, the serialized alerts by detected or created time will hide the intrinsic order between alerts. It is a disadvantage to alert fusion. In this paper, we proposed an alert fusion method based on a happened before relation, which revealed the intrinsic order between alerts. Utilizing the happened before relation can improve the performance of alert correlation and reduce the interference with other correlation components. The experiment results show that our approach is effective in achieving alert reduction and aggregation.**

Keywords-IDS; alert fusion; happened before

I. INTRODUCTION

Nowadays, IDS(intrusion detection system) has become a necessary component to network security. However, IDS can easily produce hundreds of alerts each hour, so it is necessary to find a way to reduce the amount of alerts and provides a more succinct and high-level view of security event. To address this issue, many researchers have proposed an alerts analysis process: alert correlation. The aim of alert correlation is to reduce the number of alerts and increase the relevance and abstraction level of the resulting reports. The alert correlation technology can improve the IDS performance by both achieving false positive reduction and aggregating correlated alerts into attack scenarios.

Mainly, there are three types of alert correlation techniques: multi-step, fusion-based, and filter-based. Valeur presented a general complete correlation model that includes a comprehensive set of correlation components and a framework [1, 2].

During the last years, a few multiphase analysis models have been proposed to correlate alert [3-5]. Using pre and post conditions to model and detect multi-step attacks and to provide a high-level view of the attack history, had been researched in [6-8]. Another example that uses pre and post conditions to identify causal relationships between alerts is JIGSAW[9]. A main limitation of those multi-step approaches is needed to manually define the pre and post conditions.

The fusion-based alert correlation technology is utilizing the alert similarity metric to fuse the correlative alerts. Valdes et al. presented a correlation process utilizing an alert similarity metric[3-5]. Debar and Wespi proposed a system that performs both fusion-based aggregation and correlation of alerts produced by a number of different sensors[10]. In the M2D2 model[11, 12], the authors proposed a formal description of sensor capabilities in terms of scope and positioning to determine if an alert is a false positive. In 2003, Autrel defined a similarity operator of two IDMEF alerts to aggregate and fuse alerts [13]. Fusion-based methods are good at reducing false positives, since that the false positive alerts usually have similar attributes, but they are not able to detect multi-step attacks.

The filter-based correlation approaches seek to identify the most important alerts in the alert stream. Porras discussed a mission-impact-based approach to alert prioritization and aggregation[14]. Using vulnerability analysis information to verify alert had been advocated to reduce the noise in the alert stream[15-17]. Recently, Xiao proposed an alert verification scheme based on attack classification to achieve the objectives of low cost and high efficiency of verification process[18]. The mainly limitation of those filter-based approaches is they must rely on the information about the security configuration of the protected network and the vulnerability information.

To our best knowledge, the order relation of alerts has not been in-depth researched in the field of alert correlation. In fact, the order relation of alerts from one IDS, even from the multi IDSs, always be regarded as a sequence serialized by the detected time or created time. We argued that utilizing the inbeing order relation in alerts can improve the performance of alert correlation and reduce the interference with other correlation component.

In this paper, we propose a happened before relation in alerts. The classical happened before relation is defined to determine the order of two events without using physical clocks in a distributed system[19]. The happened before relation show the intrinsic order between alerts. We use fusion-based techniques as example to investigate the effecting of the happened before relation in alert correlation, because the fusion-based techniques do not need additional human knowledge about the network, vulnerabilities, and configuration, which can also affect the performance of alert correlation.

978-1-4244-2108-4/08/$25.00

The remainder of this paper is structured as follows: section 2 presents the happened before relation and alert graph; section 3 presents the alert fusion algorithm; section 4 shows the preliminary evaluation and experiment results; finally, section 5 draws the conclusions and outlines the future works.

II. HAPPENED BEFORE RELATION AND ALERT GRAPH

We suppose that the IDS alert log is a alert sequence $a=a_1, a_2,\ldots, a_n$, where $a_i= (T_i,SIP_i,DIP_i,A_i,O_i)$ denotes that the source IP SIP_i host does a suspicious action A_i to destination IP DIP_i host at detected time T_i, and the O_i denotes other information. We artificially split a_i into two event e_i and e_i', where e_i denotes the event about alert a_i from the view of the source IP SIP_i host, and e_i' denotes the event about alert a_i from the view of the destination IP DIP_i host.

Although e_i and e_i' come from the same alert a_i and are marked the same detected time T_i, it is reasonable we assume that the really happened time of e_i is before the happened time of e_i'. However, we can not get the really happened time value of e_i and e_i', and only know the detected time from alert a_i.

All security event set of IDS alert log a is V; V_{ip}^s denotes the set of all events whose source IP are ip; V_{ip}^d denotes the set of all events whose destination IP are ip; V_{ip} denotes the set of all events whose source IP or destination IP are ip; IP denotes the set of all IP address in a.

The alert graph of a is defined $G=(V,E)$, where:

(1) If e_i and e_i' is a pair of events which are split from a same alert, then $(e_i, e'_i) \in E$;

(2) $\forall e_1,e_2 \in V_{ip}$ to any ip, if the detected time of e_1 is before the detected time e_2, and there does not exist $e \in V_{ip}$, where the detected time of e is before the detected time of e_2 and after the detected time of e_1, then $(e_1,e_2) \in E$.

The alert graph depicts the intrinsic alerts order which had been hidden by serialized alert (in Fig. 1: An example of alert graph about the three hosts).

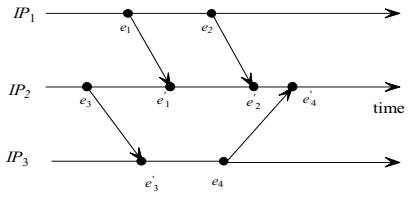

Figure 1. An example of alert graph

The definition of the "$\mapsto$" in graph G is: if $e_1,e_2 \in V$ and $(e_1, e_2) \in E$, then $e_1 \mapsto e_2$.

The definition of the "$\rightarrow$" or "happened before" relation is as follows:

(1) If $e_1 \mapsto e_2$, then $e_1 \rightarrow e_2$;

(2) If $e_1 \rightarrow e_2$ and $e_2 \rightarrow e_3$, then $e_1 \rightarrow e_3$.

Two distinct events e_1 and e_2 are concurrent $e_1 \| e_2$: if $\neg(e_1 \rightarrow e_2)$ and $\neg(e_2 \rightarrow e_1)$. Following the definition, the happened before relation is an irreflexive partial ordering relation on the set of V. We can also define the logic clock as Lamport[19].

The happened before relation is the intrinsic order between alerts. We expect that utilizing the happened before relation revivify the original order of alert, and eliminate the disadvantage of the serialized alerts by detected timestamp when alerts are fusing.

III. ALERT FUSION

Assuming $G=(V,E)$, if (Fig. 2):

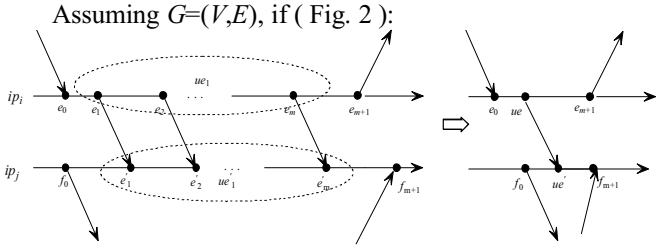

Figure 2. Alert fusion

(1) $\{e_1,e_2,\ldots,e_m\} \subseteq V_{ip_i}^s$, $\{e_0,e_{m+1}\} \subseteq V_{ip_i}$, and $e_0 \mapsto e_1 \mapsto \ldots \mapsto e_{m+1}$;

(2) $\{e_1',e_2',\ldots,e_m'\} \subseteq V_{ip_j}^d$, $\{f_0,f_{m+1}\} \subseteq V_{ip_j}$ and $f_0 \mapsto e_1' \mapsto e_2' \mapsto \ldots \mapsto e_m' \mapsto f_{m+1}$;

(3) $(e_i,e_i') \in E$ to $\forall i \in [1,2,\ldots,m]$;

then

(1) $\{e_1,e_2,\ldots,e_m\}$ and $\{e_1',e_2',\ldots,e_m'\}$ can be fused into ue and ue' respectively;

(2) $\{(e_i,e_i') \mid i \in [1,2,\ldots,m]\}$ can be fused into (ue,ue');

$G=(V,E)$ becomes $\tilde{G}=(\tilde{V},\tilde{E})$, where $\tilde{V}=V-\{e_1,e_2,\ldots,e_m\}-\{e_1',e_2',\ldots,e_m'\}+\{ue,ue'\}$, and

$$\begin{aligned}\tilde{E} = E &- \{(e_i,e_i') \mid i \in [1,2,\ldots,m]\} + \{(ue,ue')\} \\ &- \{(e_i,e_{i+1}) \mid i \in [0,2,\ldots,m]\} \\ &+ \{(e_0,ue),(ue,e_{m+1})\} \\ &- \{(e_i',e_{i+1}') \mid i \in [1,2,\ldots,m-1]\} \\ &- \{(f_0,e_1'),(e_m',f_{m+1})\} + \{(f_0,ue'),(ue',f_{m+1})\}\end{aligned}$$

Evidently, the corresponding relation $\tilde{\rightarrow}$ in $\tilde{G}$ is consistent with $\rightarrow$ in G, i.e. if $\tilde{e}_1,\tilde{e}_2$ in $\tilde{G}$ come from e_1,e_2 in G

respectively, and $e_1 \rightarrow e_2$, then $\tilde{e}_1 \tilde{\rightarrow} \tilde{e}_2$. This property can insure that the fusion operation based happened before relation can reduce the interference with other correlation component as soon as possible.

An online algorithmic of alert fusion:

Input: alert log sequence $a_1, a_2, \ldots, a_n$;

Output: reduced and aggregated super alert log sequence $a'_1, a'_2, \ldots, a'_m$;

Key array variable**:** *cut*[ ], where *cut*[*ip*] store the current alert information about the host *ip*.

(1) i=1;

(2) Draw out source IP sip_i and destination IP dip_i from a_i;

(3) **If** the source and destination IP addresses of *cut*[sip_i] equal to sip_i and dip_i respectively, and *cut*[sip_i] and a_i satisfy the *PRC* (the prior restricted conditions);**then** the *cut*[sip_i] and a_i can be fused into a super alert, and store the super alert information into *cut*[sip_i] and *cut*[dip_i];**else** { output *cut*[sip_i]; *cut*[sip_i]=a_i; *cut*[dip_i]=a_i }

(4) i=i+1; Goto (1) until the alert log is processed over.

The array variable *cut*[ ] can use aging technology by keeping a sliding time window to resolve the problem of infinite increase. In fact, the *PRC* denotes the fusing condition based on happened before relation.

IV. PRELIMINARY EVALUATION

The evaluation data set is MIT/LL 2000 data set[20], and the intrusion detection system is Snort[21] (Version 2.7.0) with default configuration and 8118 rules (download from www.snort.org at 2007-7-22). The alert fusion tool is implemented in Perl. The data set summary information is listed in Table 1. The *PRC* is listed in table 2, in which, for example, the *PRC*3 denotes that two alerts must have the same protocol, source IP, destination IP, source port, and destination port, but the priority and other information can be different.

Table 1 The data set summary information

data set	*packet*	*raw alert*
outside-tcpdump 1.0	394089	3668
inside-tcpdump 1.0	649787	1362
outside-tcpdump 2.0	236753	1392
inside-tcpdump 2.0	347987	761

When using the different PRCs, the fusion effect is different also. The detail result is described in table 3 and the average reduce ratio is depicted in Fig. 3. From table 3, we can easily find that the reducing ratio of the outside data set is larger than the inside's. The causation could be that the router between outside and inside has been filtrated much suspicious packets data. From Fig. 3, we can know that the *PRC* is looser, the reducing ratio is great. The great reducing ratio denotes the focused alert is more aggregate. Furthermore, the different *PRC*s will result in that the aggregated alerts have different meanings. Therefore, it is true to use appropriate *PRC* by actual needing.

Table 3 The alert reducing by the different *PRC*s

data set	*raw alert*	PRC*1*	PRC*2*	PRC*3*	PRC*4*	PRC*5*	PRC*6*
outside-1.0	3668	1413	1410	1378	1204	1409	1171
inside-1.0	1362	1198	1193	1179	988	1189	663
outside-2.0	1392	369	355	351	185	347	175
inside-2.0	761	750	746	740	574	738	564

For example, a super alert using *PRC*1 at outside-tcpdump 1.0 is:

Using PRC: 1
Original alert number: 254
Start time: 03/08-01:13:13.184869
End time: 03/08-01:17:07.659653
Protocol: ICMP
Source IP: 172.16.114.1
Destination IP: 172.16.114.50
Source port:
Destination port:
Priority: 2
Other information: ICMP redirect host
Classification: Potentially bad traffic

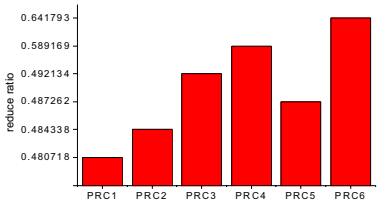

Figure 3. The average reduce ratio

Because of own limit, the number of true raw alerts detected and reported by Snort are 6 and 12 in simulative attack phase 2 (break into mill.eyrie.af.mil via the sadmind exploit)and phase 4(Initiate attack on other Eyrie hosts) on inside-tcpdump 2.0. When use *PRC*1 the number true alerts will be reduced form 6, 12 to 6, 7 in attack phase 2 and 4 respectively, but when use *PRC*3 the number of true alert will be reduced form 6, 12 to 4, 5 respectively.

We change the parameters of the aging time windows from 1s to 7200s to investigate the effecting of the output super alert number and the size of the array *cut*[] on the test data set outside-tcpdump1.0 and the prior restricted conditions *RPC*1. When the aging time windows is increased, the output alert number will be decreased, but the size of the array *cut*[] will be increased (Fig. 4). Moreover, when the aging time windows is great than 30s, the two curves trend become evenness. Setting the value of the aging time windows to ≥30s is a good choice in this case.

Table 2 The prior restricted conditions

PRC	*protocol*	*source IP*	*destination IP*	*source port*	*destination port*	*priority*	*other*
PRC_1	1	1	1	1	1	1	1
PRC_2	1	1	1	1	1	1	0
PRC_3	1	1	1	1	1	0	0
PRC_4	1	1	1	1	0	1	0
PRC_5	1	1	1	0	1	1	0
PRC_6	1	1	1	0	0	1	0

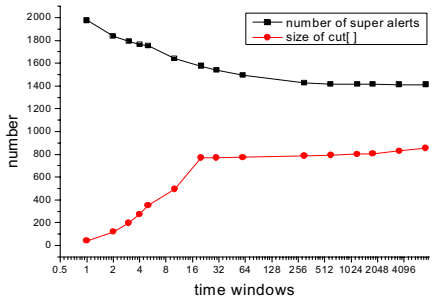

Figure 4. Impact of the aging time windows(noting: the *x*-axis is log-scaled)

Even though the goals of correlation seem to be well-defined, the correlation approaches were proposed so far emphasize different aspects of the correlation process, making it difficult to compare the results of each solution [22]. To the same test data set MIT/LL 2000, the fusion effect of the other similar methods described in [1] is presented in table 4. The attack thread reconstruction combines a series of alerts that refer to attacks launched by one attacker against a single target; but the attack focus recognition aggregates the alerts associated with single hosts attacking multiple victims (called a one2many scenario) and single victims that are targeted by multiple attackers(called a many2one scenario). From the table 4 and Fig. 3, we can know that our method and attack focus recognition have an encouraging reducing rate. By carefully analyzing, we find that the attack focus recognition resembles our method using *PRC*4 and *PRC*5, and they very effective in achieving alert reduction for that DDoS or scanning attempts. More over, our fusion method can reduce the interference with other correlation component as soon as possible by maintain the happened before relation.

Table 4 The alert reduction by the different reducing method

method	***reduction***
attack thread reconstruction	6.61%
attack focus recognition	49.58%

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed an alert fusion method based on the happened before relation. Using happened before relation, we can transform the serialized alerts by detected or created time into alert graph, which reveals the intrinsic order in alerts so that more alerts can be fused and aggregated. We used fusion-based technology to investigate the effecting of the happened before relation in alert correlation. But our method is only one component to the overall goals of alert correlation. We pursue to integrate our methods with other alert correlation technology for more effective in achieving alert reduction and aggregation. We also believe that our method can also be applied in distributed IDS environment, which is part of our ongoing works.

REFERENCES

[1] F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer, "A Comprehensive Approach to Intrusion Detection Alert Correlation", IEEE Transactions On Dependable And Secure Computing, 2004, pp. 1-23.

[2] F. Valeur, "Real-Time Intrusion Detection Alert Correlation", Doctor of Philosophy: University Of California Santa Barbara, 2006.

[3] A. Valdes and K. Skinner, "An Approach to Sensor Correlation", presented at Recent Advances in Intrusion Detection, 2000.

[4] A. Valdes and K. Skinner, "Probabilistic Alert Correlation", presented at Recent Advances in Intrusion Detection, 2001.

[5] D. Andersson, M. Fong, and A. Valdes, "Heterogeneous Sensor Correlation: A Case Study of Live Traffic Analysis", presented at Proc. Third Ann. IEEE Information Assurance Workshop, 2002.

[6] F. Cuppens and A. Miege, "Alert Correlation in a Cooperative Intrusion Detection Framework", presented at IEEE Symp. Security and Privacy, 2002.

[7] P. Ning, Y. Cui, and D. S. Reeves, "Analyzing Intensive Intrusion Alerts Via Correlation," presented at the Recent Advances in Intrusion Detection, 2002.

[8] [8] P. Ning, Y. Cui, and D. S. Reeves, "Constructing Attack Scenarios through Correlation of Intrusion Alerts", presented at ACM Conf. Computer and Comm. Security, 2002.

[9] S. J. Templeton and K. Levitt, "A Requires/Provides Model for Computer Attacks", presented at New Security Paradigms Workshop, 2000.

[10] H. Debar and A. Wespi, "Aggregation and Correlation of Intrusion Detection Alerts", presented at In Proceedings of the International Symposium on Recent Advances in Intrusion Detection, Davis, CA, 2001.

[11] B. Morin, L. Me, H. Debar, and M. Ducasse, "M2D2: A Formal Data Model for IDS Alert Correlation", presented at In Proc. of Recent Advances in Intrusion Detection, 2002.

[12] B. Morin and H. Debar, "Correlation of Intrusion Symptoms: an Application of Chronicles", presented at In Proceedings of the International Symposium on Recent Advances in Intrusion Detection, Pittsburgh, PA, 2003.

[13] F. Autrel and F. Cuppens, "Using an Intrusion Detection Alert Similarity Operator to Aggregate and Fuse Alerts", presented at the 4th Conference on Security and Network Architectures, Batz sur Mer, 2005.

[14] P. Porras, M. Fong, and A. Valdes, "A Mission-Impact-Based Approach to INFOSEC Alarm Correlation", presented at the International Symposium on the Recent Advances in Intrusion Detection, Zurich, Switzerland, 2002.

[15] R. Gula, "Correlating IDS Alerts with Vulnerability Information", 2002.

[16] N. Desai, "IDS Correlation of VA Data and IDS Alerts", 2003.

[17] C. Kruegel and W. Robertson, "Alert Verication: Determining the Success of Intrusion Attempts", presented at the 1st Workshop on the Detection of Intrusions and Malware and Vulnerability Assessment, 2004.

[18] M. Xiao and D. Xiao, "Alert Verification Based on Attack Classification in Collaborative Intrusion Detection", presented at The Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007.

[19] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System", Communications of the ACM, 1978, pp. 558-565.

[20] M. L. Laboratory, "Lincoln Lab Data Sets", 2000.

[21] R. Martin, "Snort-Lightweight intrusion detection for networks", in Proc. USENIX LISA '99 Conf., 1999.

[22] J. Haines, D. K. Ryder, L. Tinnel, and S. Taylor, "Validation of Sensor Alert Correlators", IEEE Security and Privacy Magazine, 2003, pp. 46-56.