Towards Predictive Real-time Multi-sensors Intrusion Alert Correlation Framework

Maheyzah Md Siraj^{1,2*}, Hashim Hussein Taha Albasheer¹ and Mazura Mat Din¹

¹Information Assurance and Security Research Group (IASRG), Faculty of Computing, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia; maheyzah@utm.my ²Department of Information Systems, Faculty of Computer, King Khalid University - 62529, Saudi Arabia

Abstract

Despite of Network Intrusion Detection System/Sensors (NIDSs) deployment in the computer networks to detect various attacks, it raises a serious problem. They generate a high volume of low-quality intrusion alerts when attack scenarios have taken place. Worst, NIDSs cannot extract or even predict sequence of attack scenarios. Thus, alert post-processing or known as Alert Correlation (AC) is much needed to derive current system security. AC aims to identify the complete relationship among intrusion alerts that can reveal the attacker strategy (i.e., sequence of attack scenarios). Current works do not provide attack prediction and proactive actions when attack scenarios were launched. Prediction can assists early warning and prevention to avoid the attack from escalating and damaging the network. In this paper, we highlight the important research problems in developing AC which has motivate us to propose a new AC framework design that include attack prediction and proactive step in a real-time multiple sensors environment. It is worth mentioning that to complement NIDSs in detecting the incoming attacks, intrusion alert prediction is an exploratory area for future research for the purpose of improving the quality of correlation and predicting the next attacker scenario as a proactive step.

Keywords: Attack Scenarios Prediction, Intrusion Alert Prediction, Network Security, NIDS, Real-Time Alert Correlation

1. Introduction

The growing interest in the internet, communication networks and other telecommunications alternative, with the cybercrime issues necessitated robust defensive mechanisms. Simple attacks are no longer used, so we faced every day, advanced and modern attacks exploiting our emerging services. Network Intrusion Detection System (NIDS) is a security mechanism devoted to detect attacker activities on a network. Therefore, NIDS technologies play a vital role in protecting above mentioned against cybercrime. However, these technologies are still suffered from limitation which is producing a high volume of low-quality intrusion alerts; including high rate of false positives, redundant alerts, low severity level of alerts and invalid alerts. Accordingly, such problem happens due to poor detection mechanisms used in NIDSs and therefore, produce very high rate of low-quality alerts. This makes alert analysis or NIDS post-processing very difficult and time consuming for the security analyst to manage and verify true alerts. Thus, automated alert analysis via correlation or simply called as Alert Correlation (AC) is crucial in order to reduce the number of alerts and improve the quality of alerts. An AC framework may consist of several tasks: normalization, reduction, severity/prioritization, attack detection and prediction to provide a high-level view of network security situations. The purposes of AC are to:

- Format and standardize intrusion alerts.
 - Reduce and eliminate redundant of intrusion alerts.

*Author for correspondence

- Filter low quality intrusion alerts.
- Discover attack scenario.
- Filter and prioritize intrusion alerts.
- Predict attacker next action.
- Predict forthcoming attacks.

These goals require a framework that effectively, efficiently and accurately deals with the massive alerts. Predicting the next actions of the attacker is very important and difficult task². Prediction helps intrusion prevention systems reacts properly before the network is compromised and gives the opportunities to overcome the advantages of attacker. However, existing works on overcoming the limitation of NIDSs (in term of producing high volume of low quality alerts) neither dealing with Alert Correlation^{3,4} nor Attack Prediction^{1,5} as a proactive approach.

Formally, there are many works^{1,5-13} have been proposed in AC focusing on analyzing intrusion alerts produced by NIDS to provide a concise and a high-level Figure when attack activates have taken place in timely manner. AC is a complex multi-stages transformation process and most of existing frameworks suffer from: complex correlation rules definition⁵ that limits the capabilities of detecting new attack scenarios due to hard-coded domain knowledge must be accurately predefined, depends on human expert's knowledge⁵ as well as did not provide a proactive action when attack activities going on⁵.

Most current AC11,14-16 do not predict alerts and attacks as well, they do not provide an advance warning before violation. They correlate alerts to detect the attack pattern for forensics purpose; there are few works^{1,5,13,17} to predict attack but they do not look for the AC as Alert Prediction problem. As such these ACs are often concentrated to the role of post analysis rather than being proactive^{2,18}. As well, by prediction false alerts can be excluded because they are often of isolated and non-critical events, and therefore, this can help in terms of quality. In this paper, we focus on anticipate and conduct possible attacks to prevent damage as proactive step as well to help in improving the quality of alerts and correlation process. Our proposed framework aims toward a predictive yet effective, real-time multi-sensors intrusion alert correlation.

The rest of this paper is organized as follows. Section 2 surveys the related works on AC. Section 3 summarizes the comparison on existing alert correlation frameworks based on their approaches. The proposed AC framework

design is presented in Section 4. Finally, Section 5 concludes the paper and outlines achievable future works.

2. Related Works

Alert post-processing or Alert Correlation (AC) is much needed in order to improve the quality alerts that have been generated by NIDSs. Based on previous works, we generalized the components in AC framework into five: Normalization/Formatting, Reduction, Severity/ Prioritization, Attack Scenario Construction, and Attack Prediction. The following sections explain more detail for each of them.

2.1 Alert Normalization/Formatting

Formatting the alerts can be considered as an important initial task in the preprocessing task of AC framework. Nowadays, the majority of organizations implement different types of NIDSs (heterogeneous NIDSs), accordingly they produce alerts in different data format. Alert normalization is a process to convert different alert data formats from multiple intrusion sensors into a standard format to be appropriate and acceptable by the other correlation components.

Debar and Wespi¹⁹ are among the earliest researchers that addressed the problem of formatting and standardizing the unformatted alerts. Their work has motivated the IETF Intrusion Detection Working Group (IDWG)²⁰ to design a standard format known as Intrusion Detection Message Exchange Format (IDMEF) that can be adopted in all types of IDSs. IDMEF data model is a standard representation of alerts into a class with the following set of attributes: {Alert ID, Sensor ID, Timestamp, Source IP Address, Source Port, Destination IP Address, Destination Port, Service Protocol, Alert Type}⁴. IDMEF has been applied in many AC works for examples in²¹ and now IDMF is only reliable format applied by alert correlation researchers¹⁵.

But, IDMEF cannot address the problem of alert redundancy where NIDSs produced multiple repeated (similar) alerts in a short duration of time. Such redundant alerts can overload and may contribute to false correlation. Therefore, they must be identified and reduced.

2.2 Alert Reduction

NIDSs can easily trigger thousands of alerts per day, this

flow contains repeated/redundant, low interesting and up to 99% are false positives alerts^{11,14,22-25}. In order to reduce the alerts, we categorized the related works in two: Aggregation and Filtration.

2.2.1 Alert Aggregation

Alerts that are generated by same NIDS or different NIDSs usually belong to the same attack. They are identified by the same source and target IP addresses and blended with repeated/redundant alerts²⁶. Such case increased the number of alerts and produce high-dimensionality of alerts. In practical, the redundant alerts are usually false positives²⁴.

Aggregation is used to group repeated/redundant alerts and will be represented as one meta-alert or hyperalert¹⁹. Alerts are aggregated (or clustered) based on feature/attribute similarities as proposed by ²⁴ and ¹⁹. The similarities are identified using either predefined rules ¹⁴ or similarity operator/function^{7,8,27}. Each cluster is then merged and a new, global alert is generated to represent the whole cluster^{3,7,12,28}. Aggregation is practical since the similarity searching among alerts can be done automatically on a high number of alerts.

Another effort to reduce the redundancy alerts is through a combination of a throttling algorithm namely a Token Bucket Filter (TBF) with an existing alert correlation model²⁹. However, this process requires much human efforts and time. Another effort attempts to reduce unnecessary alerts by validating alerts with vulnerabilities assessment and then aggregate alerts^{16,30,31}. Kavousi and Akbari⁵ aggregate alerts that have been triggered for same attack step. Zomlot³² used Support Vector Matrix (SVM) to reduce alerts, and the non-interested alert is not removed, they claim it will be helpful to link true attacks. However, reducing redundant alerts do not truly eliminate false positives¹⁷. Therefore, the next problem is to verify and filter (remove) false positive alerts.

2.2.2 Alert Filtration

Technically, false positives alerts are caused by³³:

- Runtime limitations.
- Specificity of detection signatures.
- Dependency on environment. Actions that are normal in certain environments may be malicious in others.

Therefore, to produce effective correlation on the alerts, the false positives have to be verified and filtered. A more practical solution to filter and reduce the number of

false positives is by processing and correlating the alerts. Maggi and Zanero³⁴ have used machine learning to filter out the false positives, ^{35–37} have adopted a fuzzy-based classifier to generate fuzzy rules that classified alerts as true or false positives using background knowledge, while ³⁸ have used genetic based fuzzy classifier to generate the classification rules. Whereas in ³⁹, a classical clustering algorithm based on XML distance measure to group the alert patterns into clusters is implemented. Each XML document represents a pattern of alerts for a network session.

Xiao and others¹¹ have introduced outlier detection data mining technique to identifying true alert and reducing false positive, their method also can learn from these alerts and automatically adjust the filtering mechanism to new situations.

Instead of treated the false positives as a filtering problem, Yu and Frincke¹⁷ perceived it as an inference problem. They made an assumption that the intruder's actions are unknown to the NIDS and can be inferred only from the alerts generated by the NIDSs. The model is known as Hidden Colored Petri-Net (HCPN). HCPN can describe the relationship between different steps carried out by intruders, model observations (alerts) and transitions (actions) separately, and associate each token element (system state) with a probability (or confidence). However, this approach is effective on known alerts only^{36,37}. Since the current AC framework should cover the known and new alerts, thus machine learning approach is more suitable for dealing with false positives due to its capability of learning and training in recognizing new alerts.

Although the filtration of false positives and aggregation of repeated alerts can improve the alerts quality, the hidden useful meaning contains in the alerts is still unrevealed. Thus, extraction of meaningful information from the alerts can be achieved by recognizing the attack scenario.

2.3 Alert Severity/Prioritization

Not all generated alerts are equally important in term of their severity and critically of the target being attacked²⁵, so there is need to separate few important alerts from the rest and give them priority. Work by ⁴⁰ categorized severity of alerts into three types: low, medium and high. High severity alerts are always referred to high risk alerts that can cause huge damage to network assets. They used infor-

mation in the NIDS signature files to identify the type of severity. Normally, alerts that are low severity will be ignored by security analyst for future correlation process. Alsubhi et al.⁴¹ proposed a fuzzy-logic based technique for scoring and prioritizing alerts. Their method evaluates alerts based on a number of criteria and used Fuzzy logic inference mechanism in order to score/prioritize alerts.

2.4 Attack Scenario Construction

Constructing attack scenario is important and crucial in AC research^{11,13,25} to study the behavior of the attacker^{13,42,43}. It is challenging because alerts contain low level information¹⁵. In practical, attack scenario should consist of a number of attack stages, and an attack stage should contain a list of attack steps. Therefore, in order to recognize attack scenario, two problems need to be addressed: 1) Identifying Attack Steps, and 2) Recognizing Attack Stages.

Common pattern of alerts should bring useful information. Finding the commonalities among group of alerts is the problem of identifying the common attack steps. This problem can be solved by clustering/grouping common alerts based on the similarities of certain or all attributes⁴⁴. They are two issues need to be considered: i) How to define and determine the level of similarity, and ii) How to group unknown/new alerts. In determining level of similarity, some researchers for example^{26,45} used a predefined similarity probabilistic-based function to measure similarity between two alerts. Dain and Cunningham⁴⁶ required a number of predetermined rules defined by security analyst to group alerts. Alerts that fulfilled the rules of an attack step should belong in a group. But, their works partially addressed this problem since rules for new alerts are unpredictable¹⁹. Thus, new alerts are isolated and cannot be correlated.

Grouping the unknown or new alerts can be achieved by using unsupervised machine learning algorithms^{3,44,47}. They have shown that grouping similar attributes not only reveals the attack steps, but it can reduce a large a number of alerts as well. Even though clustering can effectively correlate some alerts, it cannot discover the causal relationships between alerts. Because of that, recognizing attack stages is essential to discover the causal relationships.

Recognizing attack stages are closely related to a classification problem because it attempts to classify the alerts into the corresponding cause/class. Based on the

cause-effect paradigm, ^{8,42,48,49} derived rules and knowledge on the known attack stages to construct the 'cause' and 'effect' of an attack stage. As the patterns of intrusion changes, the classification should flexible enough to permit the introduction of new alerts where their properties may belong to neither class nor several classes. The above works cannot handle such cases. That is why training-testing paradigm using supervised machine learning algorithms^{9,33,39,50,51} is more practical. But, if using unlabeled dataset, the labeling of target attribute for data training needs to be done beforehand.

2.5 Attack Prediction

As well known, NIDS technologies play a vital role in protecting communication networks against cybercrime. However, these technologies are not very effective in predicting the future attacks⁵². Worst, it generates alerts when attack activities/intrusions have taken place. A proactive approach is to anticipate and conduct possible attacks to prevent damage. Accordingly, the next step of an attack can be predicted after detection of few steps of attack in progress³¹, so predicting the next actions of the attackers is an important and difficult task^{1,2,53,54}. Attack Prediction can help intrusion prevention systems reacting properly before the network is compromised having the opportunities to overcome the advantages of attacker.

3. Comparison on Existing Alert Correlation Frameworks based on Approach Used

NIDS post-processing has been studied for more than 10 years to overcome the limitation of NIDS, especially high volume of low quality alerts. Up to now, a variety of alert correlation approaches have been proposed. The existing approaches can be categorized as similarity-based, statistical-based, knowledge-based alert correlation^{2,21,55,56}. In addition to these approaches here we present a hybrid-based approach. Briefly, similarity-based approach are focused on addressing the issue of improving the quality of alerts at reducing alerts based on the similarities between alert attributes; whereas statistical-based approach are dealing with the issue of recognizing the attack scenario based on statistical or casual relationship between alerts; knowledge-based approach are dealing with attack defini-

Approach	Pre-knowledge or rules	Alert Reduction	Reducing False Alert	Alert Prioritization	Extract Attack Scenario	Predict Next Alert	Construct and Predict attack scenario	Shortcoming points
Similarity-based	~	~	√		~			-Suitable for known alerts. -Not apple to discover causality of alerts and statistical relationships. -Limited to discover complicated attacks.
Statistical-based	~	\checkmark	~	~	~	\checkmark		-Not able to discover dependences. -Difficult to estimate correlation parameters. -Not able to discover structure and means/ cause relationships.
Knowledge-based	\checkmark			~	✓			 Need manually define prerequisites. Not able to deal with new pattern of alerts. Difficult to update the correlation knowledge. Not able to discover structure and statistical relationships. Impractical for use in large scale or real time due computational side.
Hybrid-based	\checkmark	\checkmark	V	~	~			-May lead to complex architecture.

Table 1. Comparison on alert correlation approaches

tion based on alert meaning. Last not least, a hybrid-based approach attempt to exploit the strengths of each of the three correlation approaches. Brief comparison for these approaches is presented in Table 1.

4. The Proposed Predictive Real-time Alert Correlation Framework

Attack prediction is an important capability and difficult task^{1,2,5,18,57}, it completes the role of NIDSs as systems that predict future hacker actions before damage, and automatically respond to attacks in a timely manner.

To proactive intrusion must be able to infer the goals of attackers. Identifying the attacks is not enough, so we

need to understand the plan of the attacker, and predict the next actions. After predict the next attacker event/ action, easily we can complete the high level picture of the attack.

Figure 1 illustrate the design of the proposed alert correlation framework, which consist of three main components, they are:

- 1) Online alert preprocessing with early correlation process,
- 2) Online predictive alert correlator, and
- 3) Online and offline alert optimizer.

The proposed framework aims to achieve the following objectives:

- Handling multi sensors and real time intrusion alerts
- Formatting and standardizing intrusion alerts.

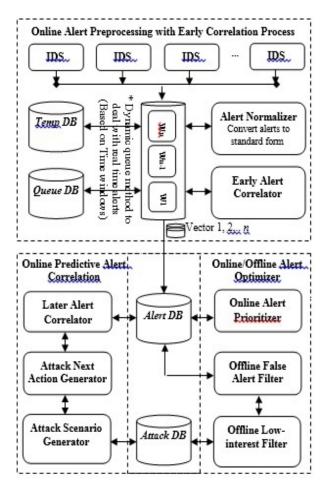


Figure 1. The proposed framework.

- Reducing and eliminating redundant of intrusion alerts.
- Filtering low-interest and false positive intrusion alerts.
- Discovering attack scenario.
- Filtering and prioritizing intrusion alerts.
- Forecasting attacker next action.
- Forecasting forthcoming attacks.

With automated correlation framework, the following sub-sections explain how to gain aforementioned objectives.

4.1 Online Alert Preprocessing with Early Correlation Process

Basically, this component contains three processes and two databases as shown in Figure 2, which they works together to achieve the following objectives:

Handle multi-sensors IDS alerts in real-time.

- Convert intrusion alerts to standard format (e.g. IDMEF).
- Improve the quality of alerts (Filtering redundant and invalid alerts).
- Detect known attacks.
- Produce vectors of alerts (e.g. based on similarities among alerts).

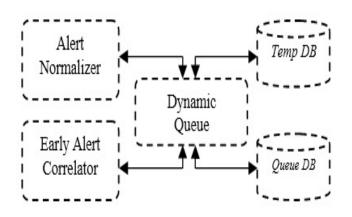


Figure 2. The main units of online alert preprocessing with early correlation process.

4.1.1 Alert Normalizer

As aforementioned, majority of organizations implement different products of NIDSs (Multi-Sensor), and they produce alert set in different data format. These incoming intrusion alerts are dynamic and collected continuously in time, so the real-time extension is an efficient approach for the framework to be proactive.

This component efficiently aggregate real-time multisensors intrusion alerts to correlate them in a real-time. Accordingly, the need for a component that is now viable in multi-sensor real-time IDS. Raw alerts are resulted in a real-time from multiple NIDSs containing all features can be correlated directly based on given feature such as time, IP source, destination address, etc.

4.1.2 Early Alert Correlator

The purpose of this unit is to reduce the number of alerts by removing low quality alerts (redundant and invalid alerts).

4.1.3 Dynamic Queue

The dynamic queue method treat live alert receiving and it is similar to the first in first our principle in a given time window (e.g. 3 sec). Received alerts identified by a uniqueID based on Intrusion Detection Message Exchange Format (IDMEF) which is commonly used to model security alerts. In this early stage alerts are correlated as explain below which lead to construct a connection vector in the current window.

The proposed queue uses a database to store all incoming alerts from IDS sensors to correlate them (Early Correlation - filter out redundant duplicated alerts), a temporal database is used to store uncorrelated alerts.

However, the correlation of some features do not rely on the time slot window wi, but on the number of the same or different alerts in previous time, e.g. last 1000 alerts we keep it as backlog to precisely correlate them. Finally, all windows $w_1, w_2,..., w_1$ inside the queue will contain a certain number of alert vectors $V_1, V_2,..., V_n$ which generated from the early correlation process.

4.2 Online Predictive Alert Correlation

As well, this component contain three process and two databases as shown in Figure 3, which they works together to improve the quality of alerts by achieving the following objectives:

- Detect unknown attacks.
- Discovering attack scenario.
- Predict attacker next action.
- Predict forthcoming attacks.

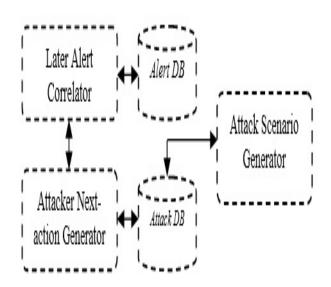


Figure 3. The main units of online predictive alert correlation.

4.2.1 Later Alert Correlator

The purpose of this unit is to identify and reveal optimal correlation among alerts, and this is complementary of the correlation process.

4.2.2 Attacker Next-action Generator

Identifying the attacks (known or unknown) is not enough. In this process, the aim is to infer attacker next action as a proactive step. This will help Attacker Scenario Generator to construct all possible alert pattern may cause attack in future.

4.2.3 Attack Scenario Generator

After predict the next attacker event/action, easily we can complete the high level picture of the attack. In this process, the aim is to construct all possible attack scenarios (known and unknown) from alert base and attack base. This will help Online and Offline Alert Optimizer to improve the quality of alerts in filtering low-interest (alerts that not completing the attack scenario) and false alerts.

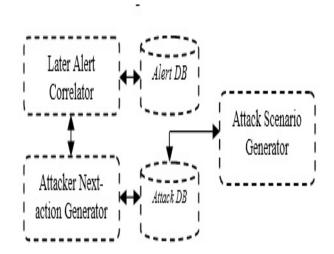


Figure 4. The main units of online and offline alert optimizer.

4.3 Online and Offline Alert Optimizer

As well, this component contain three process and two database as shown in Figure 4, which they works together to improve the quality of alerts by achieving the following objectives:

- Prioritizing intrusion alerts.
- Filtering false positive intrusion alerts.

• Filtering low-interest intrusion alerts.

5. Conclusion

In this paper, we propose a solution strategy direction for NIDS post-processing problem as Attack Prediction based on the philosophy intrusion prediction is an important technique to help response systems react before the network is compromised, and to have the opportunities to overcome the advantages of attacker by predicting the next attacker action as a proactive step. This direction is different from existing works; our method deal with AC problem as prediction problem, notably produced alerts as Relational Time Series. Online and offline alert optimizer helps up better in term of correctness. In future, we will consider this direction in several aspects: 1) validate the early correlation process for effectiveness and performance, 2) study on different types of relation time series prediction algorithms for attack prediction, and 3) study on different types of dataset such as DARPA2000 and Koyto2006+ to measure correlation effectiveness and completeness of the proposed framework.

6. Acknowledgement

We would like to thank Ministry of Education (MoE) and Universiti Teknologi Malaysia for funding this work under vot number (02G62).

7. References

- Cipriano C, Zand A, Houmansadr A, Kruegel C, Vigna G. Nexat: A history-based approach to predict attacker actions. Proceedings of the 27th Annual Computer Security Applications Conference; 2011. p. 383–92.
- 2. Yu Beng L, Ramadass S, Manickam S, Soo FT. A Survey of intrusion alert correlation and its design considerations. IETE Technical Review. 2014; 31(3):233–40.
- 3. Man D, Yang W, Wang W, Xuan S. An alert aggregation algorithm based on iterative self-organization. Procedia Engineering. 2012; 29:3033–8.
- 4. Elshoush HT, Osman IM. An improved framework for intrusion alert correlation. Proceedings of the World Congress on Engineering; 2012.
- 5. Kavousi F, Akbari B. A Bayesian network-based approach for learning attack strategies from intrusion alerts. Security and Communication Networks. 2014; 7(5):833–53.

- 6. Bloedorn E, Hill B, Christiansen A, Skorupka C, Talbot L, Tivel J. Data mining for improving intrusion detection. Ed: Technical report, MITRE. 2000.
- Cuppens F. Managing alerts in a multi-intrusion detection environment. Computer Security Applications Conference; 2001. p. 22.
- Cuppens F, Miege A. Alert correlation in a cooperative intrusion detection framework. Proceedings of IEEE Symposium on Security and Privacy; 2002. p. 202–15.
- 9. Zhu B, Ghorbani AA. Alert correlation for extracting attack strategies. 2005. Citeseer.
- Zan X, Gao F, Han J, Sun Y. A Hidden Markov Model based framework for tracking and predicting of attack intention. International Conference on Multimedia Information Networking and Security, MINES'09. 2009; 2:498–501.
- Xiao F, Jin S, Li X. A novel data mining-based method for alert reduction and analysis. Journal of Networks. 2010; 5:88–97.
- Spathoulas GP, Katsikas SK. Enhancing IDS performance through comprehensive alert post-processing. Computers & Security. 2013; 37:176–96.
- 13. Ning P, Xu D. Learning attack strategies from intrusion alerts. Proceedings of the 10th ACM Conference on Computer and Communications security; 2003. p. 200–9.
- Julisch K, Dacier M. Mining intrusion detection alarms for actionable knowledge. Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2002. p. 366–75.
- Valeur F, Vigna G, Kruegel C, Kemmerer RA. Comprehensive approach to intrusion detection alert correlation. IEEE Transactions on Dependable and Secure Computing. 2004; 1(3):146–69.
- Sadighian A, Fernandez JM, Lemay A, Zargar ST. ONTIDS: A Highly Flexible Context-Aware and Ontology-Based Alert Correlation Framework. Foundations and Practice of Security, ed: Springer; 2014. p. 161–77.
- Yu D, Frincke D. Improving the quality of alerts and predicting intruder's next goal with Hidden Colored Petri-Net. Computer Networks. 2007; 51:632–54.
- Geib CW, Goldman RP. Plan recognition in intrusion detection systems. Proceedings of DARPA Information Survivability Conference & Exposition II. DISCEX'01. 2001; 1:46–55.
- Debar H, Wespi A. Aggregation and correlation of intrusion-detection alerts. Recent Advances in Intrusion Detection. 2001. p. 85–103.
- IETF: The Intrusion Detection Message Exchange Format [Internet]. 2004. Available from: http://tools.ietf.org/html/ draft-ietf-idwg-idmef-xml-11

- Salah S, Macia-Fernandez G, Diaz-Verdejo JE. A modelbased survey of alert correlation techniques. Computer Networks. 2013; 57(5):1289–317.
- 22. Axelsson S. The base-rate fallacy and its implications for the difficulty of intrusion detection. Proceedings of the 6th ACM Conference on Computer and Communications Security. 1999. p. 1–7.
- Clifton C, Gengo G. Developing custom intrusion detection filters using data mining. Proceedings of 21st Century Military Communications Conference. 2000; 1:440–3.
- 24. Julisch K. Mining alarm clusters to improve alarm handling efficiency. Proceedings of 17th Annual Computer Security Applications Conference. ACSAC 2001. 2001. p. 12–21.
- Ghorbani AA, Lu W, Tavallaee M. Alert management and correlation. Network Intrusion Detection and Prevention. 2010; 47:129–60.
- 26. Valdes A, Skinner K. Probabilistic alert correlation. Recent Advances in Intrusion Detection. 2001. p. 54–68.
- 27. Autrel F. Using an intrusion detection alert similarity operator to aggregate and fuse alerts. 2005.
- 28. Porras PA, Fong MW, Valdes A. A mission-impact-based approach to INFOSEC alarm correlation. Recent Advances in Intrusion Detection. 2002. p. 95–114.
- 29. Turner JS. New directions in communications (or which way to the information age?). IEEE Communications Magazine. 2002; 40(5):50–7.
- Nguyen TH, Luo J, Njogu HW. An efficient approach to reduce alerts generated by multiple IDS products. International Journal of Network Management. 2014; 24(3):153–80.
- Alserhani F. A framework for multi-stage attack detection. Proceedings of Electronics, Communications and Photonics Conference (SIECPC); 2013. p. 1–6.
- 32. Zomlot L, Chandran S, Caragea D, Ou X. Aiding intrusion analysis using machine learning. Proceedings of 12th International Conference on Machine Learning and Applications (ICMLA); 2013. p. 40–7.
- Pietraszek T, Tanner A. Data mining and machine learning towards reducing false positives in intrusion detection. Information Security Technical Report. 2005; 10(3):169– 83.
- Maggi F, Zanero S. On the use of different statistical tests for alert correlation-short paper. Recent Advances in Intrusion Detection. 2007. p. 167–77.
- Huang C-J, Hu K-W, Cheng H, Chang T-K, Luo Y-C, Lien Y-J. Application of type-2 fuzzy logic to rule-based intrusion alert correlation detection. Int J Innov Computing Inform and Control. 2012; 8(4):2865–74.
- Maggi F, Matteucci M, Zanero S. Reducing false positives in anomaly detectors through fuzzy alert aggregation. Information Fusion. 2009; 10(4):300–11.

- Alshammari R, Sonamthiang S, Teimouri M, Riordan D. Using neuro-fuzzy approach to reduce false positive alerts. Proceedings of Fifth Annual Conference on Communication Networks and Services Research. CNSR'07; 2007. p. 345–9.
- Hassan MMM, Baruah HK. Fuzzy classifier for ids alerts using genetic algorithm. International Journal of Research in Advent Technology. 2014; 2(1):228–38.
- 39. Long J, Schwartz D, Stoecklin S. Distinguishing false from true alerts in Snort by data mining patterns of alerts. Defense and Security Symposium; 2006. p. 62410B-62410B-10.
- Siraj MM. Hybrid of structural-causal and statistical model for intrusion alert correlation [PhD thesis]. Skudai: Universiti Teknologi Malaysia; 2013.
- Alsubhi K, Al-Shaer E, Boutaba R. Alert prioritization in intrusion detection systems. Proceedings of IEEE Network Operations and Management Symposium. NOMS 2008; 2008. p. 33–40.
- 42. Ning P, Cui Y, Reeves DS, Xu D. Techniques and tools for analyzing intrusion alerts. ACM Transactions on Information and System Security (TISSEC). 2004; 7(2):274–318.
- 43. Bateni M, Baraani A, Ghorbani A. Alert correlation using artificial immune recognition system. International Journal of Bio-Inspired Computation. 2012; 4(3):181–95.
- 44. Smith R, Japkowicz N, Dondo M, Mason P. Using unsupervised learning for network alert correlation. Advances in Artificial Intelligence. 2008. p. 308–19.
- Lee S, Chung B, Kim H, Lee Y, Park C, Yoon H. Realtime analysis of intrusion detection alerts via correlation. Computers & Security. 2006; 25(3):169–183.
- 46. Dain O, Cunningham RK. Fusing a heterogeneous alert stream into scenarios. Proceedings of the 2001 ACM workshop on Data Mining for Security Applications; 2001.
- Marchetti M, Colajanni M, Manganiello F. Framework and models for multistep attack detection. International Journal of Security and Its Applications. 2011; 5(4):73–90.
- 48. Templeton SJ, Levitt K. A requires/provides model for computer attacks. Proceedings of the 2000 workshop on New security paradigms; 2001. p. 31–8.
- 49. Cuppens F, Ortalo R. LAMBDA: A language to model a database for detection of attacks. Recent Advances in Intrusion Detection. 2000. p. 197–216.
- 50. Qin X. A probabilistic-based framework for infosec alert correlation [PhD thesis]. USA: College of Computing Georgia Institute of Technology; 2005.
- Pietraszek T. Using adaptive alert classification to reduce false positives in intrusion detection. Recent Advances in Intrusion Detection. 2004; 3324:102–24.
- 52. Chintabathina S, Villacis J, Walker J, Gomez H. Plan recognition in intrusion detection systems using logic programming. Proceedings of 2012 IEEE Conference on

Technologies for Homeland Security (HST); 2012. p. 609–13.

- 53. Khong FW. Performance assessment of network intrusionalert prediction. DTIC Document. 2012.
- 54. Shameli Sendi A, Dagenais M, Jabbarifar M, Couture M. Real time intrusion prediction based on optimized alerts with Hidden Markov model. Journal of Networks. 2012; 7(2):311–21.
- 55. Sadoddin R, Ghorbani A. Alert correlation survey: framework and techniques. Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services; 2006.
- 56. Mirheidari SA, Arshad S, Jalili R. Alert Correlation Algorithms: A Survey and Taxonomy. Cyberspace Safety and Security. 2013; 8300:183–97.