NETWORK INTRUSION ALERT CORRELATION CHALLENGES AND TECHNIQUES

Maheyzah Md Siraj, Siti Zaiton Mohd Hashim

Faculty of Computer Science and Information System,
Universiti Teknologi Malaysia,
81310 Skudai, Johor, Malaysia.

Email : {maheyzah, sitizaiton}@utm.my

Abstract: Many organizations implement Intrusion Detection Systems (IDS) as the first line of defense for their security systems. Up to now, the researchers have developed IDS in many computer environments. Having detected the signs of intrusions, IDS trigger alerts to report them. These alerts are presented to human analyst to be evaluated and initiates adequate responses. But, manually analyzing those alerts are tedious, time-consuming and error-prone. The reasons for this: (1) the number of alerts is enormous, and (2) most of them are false alerts. A promising method to automate the alert analysis is finding the correlation between alerts, and such system is known as Alert Correlation System (ACS). One of the major applications of alert correlation (AC) is attack diagnosis. Interestingly, researchers have different kind of views to define the concept of AC. Furthermore, a various types of techniques have been proposed in AC: (1) to reduce the false alerts, and (2) to find causality relationship between alerts to extract the strategies of attacker. This paper discussed the challenges of ACS and the most importantly presents a review of techniques and solutions proposed in the course of the last ten years, while comparing their advantages and limitations. The survey is followed by the presentation of potential future research directions in this area.

Keywords: information assurance and security, alert correlation, alert analysis, intrusion detection, preventive system.

1. INTRODUCTION

Intrusion Detection Systems (IDS) have been extensively used by researchers and practitioners to maintain trustworthiness in computer environments [1,2]. A major problem in the IDS is the guarantee for the intrusion detection. This is the reason why in many cases IDS are used together with a human expert (or analyst). In this way, IDS is actually helping the human expert and it is not reliable enough to be trusted on its own. Among the limitations of current IDS that lead to the research of alert correlation (AC) include:

1) *IDS generates a huge number of alerts* [3-9]. For example, in [8] "five IDS sensors reported 40MB of alert data within ten days, and a large fraction of these alerts are false positives". This problem is acknowledged by practioners [2] that "IDSs trigger thousands of alerts per day".
2) *IDS cannot ensure that all alerts reflect actual attacks.* The 'true positives' (i.e., attacks detected as intrusive) alerts are usually mixed with 'false positives' alerts (i.e., benign activities detected as intrusive) [10-13]. Worse, only 1% of the enormous amount of IDS alerts corresponds to unique attacks. The remainders are false positives (i.e., alerts on non-intrusive actions), repeated warnings for the same attack, or alert notifications arising from erroneous activity or configuration artifacts [8,9].
3) *IDS alerts bring low-level information* [13]. This make human analyst hardly to (directly) understand the security state of the protected network and study the attack strategies [14].

Realizing of this shortcoming, IDS will be upgraded to Intrusion Detection and Response System (IDRS)[1]. Briefly, as shown in Figure 1, once the intrusion is detected, IDS will raise alerts independently (for each attack), though there may be logical connections between them. These alerts will be processed and analyzed using Alert Correlation System (ACS) to facilitate the Intrusive Response System (IRS). IRS is responsible to plan and develop effective response mechanisms. Automation of ACS is very important not only because to achieve a reliable and accurate response plans, but to reveal the continuously changing attack strategies. On the other hand, details on the taxonomy of IRS can be found in [17], and also its survey in [18].

This paper is focused on the second part of IDRS, the ACS. In specific, Section 2 defines the fundamental concept and its conceptual framework. Section 3 discusses the AC issues and problems faced by researchers. Section 4 explores the existing AC approaches and techniques. This is followed by Section 5 which presents comparative analysis of existing ACSs while discussing their advantages and drawbacks. Section 6 proposes open research problems that are worth to look into in the area of AC. Finally, Section 7 concluded this paper.

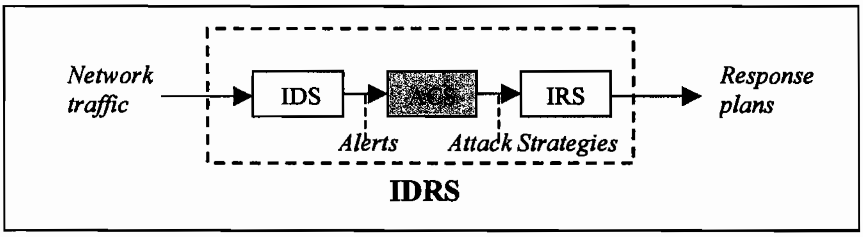

Figure 1. An Intrusion Detection and Response System (IDRS).

2. ALERT CORRELATION

A promising technique to analyze intrusion detection (ID) alerts is called *Alert Correlation* [10, 15,16]. In reality, ID community has brought the idea of managing alerts from the network surveillance where alert correlation is first used in order to reduce the workload of the analysts [19]. Computer networks produce thousands of alerts per day that make the task of the network surveillance and alert management difficult [20]. Based on this observation and the actual need for alert analysis and assessment, [19] introduced AC to cope with this phenomenon. Any tool or software system that based on the AC processes is known as Alert Correlation System (ACS).

First, let we define *correlation*. In the field of mathematics and statistics, correlation is a simultaneous change in value of two numerically valued random variables [21]. The positive correlation between cigarette smoking and the incidence of lung cancer for instance. Such statistical correlation is not appropriate for ID, since the researchers do not model alert streams by means of random variables. Correlation in analyzing alerts, however, has no consistent definition and has been used differently by researchers and IDS vendors [12]. Other definition of correlation based on a dictionary by [21] is a causal, complementary, parallel, or reciprocal relationship, especially a structural, functional, or qualitative correspondence between two comparable entities, for example a correlation between drug abuse and crime. However this definition is not very precise [12].

Precisely, the correlation in the field of alert analysis concerns finding relationships between alerts generated by a single (or multiple) data sources and coupling this information with additional knowledge [12]. It is a process that analyzes the alerts produced by one or more IDSs and provides a more succinct and high-level view of occurring or attempted intrusions [13]. AC is a challenging task in security management. Moreover, an ACS should effectively reduce the alert redundancy, intelligently analyze alerts and correctly identify the attack strategies. In fact, the goal of correlation would be to present the information generated by IDS in a meaningful way, i.e., a way which can be easily understood and processed by a human analyst and helping him/her discover attacks and incidents. It is important to notice that the goal of correlation tasks is to help the human operators as much as possible, but not to replace them.

2.1 Conceptual Framework

The ACS framework consists of several operations. Generally, the operations for an ACS framework in accordance with the literature reviewed in this research area are illustrated in Figure 2. Refering to Figure 2, in the *Preprocessing* phase, raw alerts are collected from one or many IDSs and converted to a standard format (or known as normalization), called Intrusion Detection Message Exchange Format (IDMEF)[22]. The alerts with all its attributes are then stored in a Database Management System (DBMS). To ensure the alerts are true positives, verification which is defined as the process of verifying the success of attacks is essential. Proposed by [23], it is done by comparing the configuration of the victim machine (e.g., operating system, running services, and service version) to the requirements for a successful attack. When the victim is not vulnerable to a particular attack (because the configuration does not satisfy the attack requirements), then the alert can be tagged as failed.

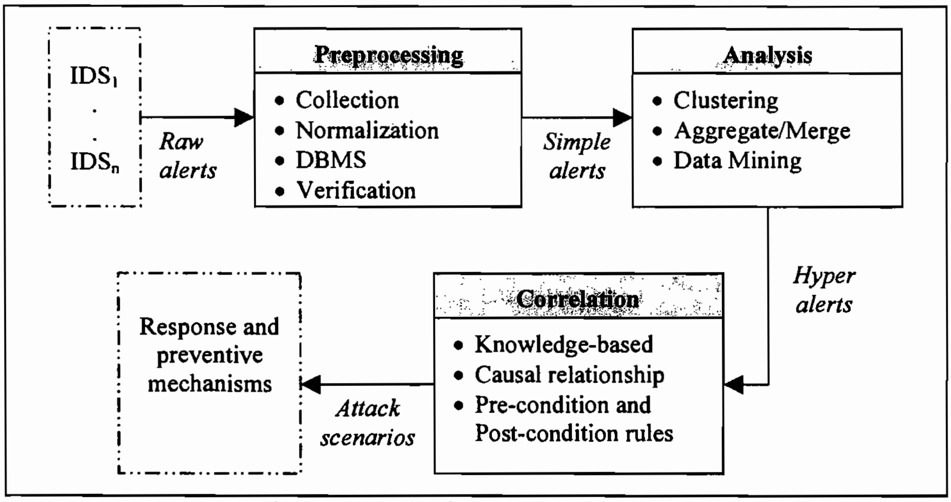

Figure 2. The framework of Alert Correlation System (ACS).

The main goal of *Analysis* phase is usually to get rid of most of the false alerts, not forgetting to merge the redundant alerts. To do this, similarity-based clustering [4] can detect redundancies and emerge the cluster as hyper alerts. Whilst in [8], researcher groups alerts based on the same root causes. The future load of alerts could be reduced by removing the most predominant and persistent root causes.

For *Correlation*, a common approach is defining rules or knowledge to determine the relationship between two alerts, as proposed in [4,7,24,25]. They predefined set of pre-condition and post-condition rules for each alerts, based on the observation that most attacks consists of many stages. Thus, alert whose post-conditions contribute to pre-conditions of the next alert can be correlated as previous alert preparing for the next one. These relationships construct attack scenarios and reveal the next goals of attacker (i.e., via observation of attack graphs). In a result, response and preventive mechanisms can be planned and developed to avoid the damage from escalating and to stop the attack at earlier stage.

3. CHALLENGES AND PROBLEMS

We classified four categories of main research issues and presented its related problems in the research of AC. They are in the following:

1) *Improving Quality of Alerts* (reducing redundant and false positives alerts thru alert post-processing),
2) *Constructing Attack Scenarios,*
3) *Extracting Attack Strategies to Predict Attacker's Next Goal*, and
4) *Other Issues.*

3.1 Issue 1 : Improving Quality of Alerts

It is known that current IDSs generate massive amount of alerts, not only because of the redundant alerts, but also the mixing of false alerts (i.e., false positives and/or false negatives). This situation contributes to low quality alerts. In fact, it has been estimated that up to 99% of alerts reported by IDSs are not related to security issues [8,9,26]. Reasons for this include the following as mentioned in [27]:

Runtime limitations : In many cases an intrusion differs only slightly from normal activities, sometimes even only the context in which the activity occurs determines whether it is intrusive. Owing to harsh real-time requirements, IDSs cannot analyze the context of all activities to the extent required [28].

Specificity of detection signatures : Writing signatures for IDSs is a very difficult task [29]. In some cases, the right balance between an overly specific signature (which is not able to capture all attacks or their variations) and an overly general one (which recognizes legitimate actions as intrusions) can be difficult to determine.

Dependency on environment : Actions that are normal in certain environments may be malicious in others [30]. For example, performing a network scan is malicious unless the computer performing it has been authorized to do so. IDSs deployed with the standard out-of-the-box configuration will most likely identify many normal activities as malicious.

As a result, the problem of false positives is critical in ID and has received considerable attention from researchers as well as practitioners [12] in different levels. There are four different levels of researches (as illustrated in Figure 3 [12]) that aim to resolve the false positives problem in the area of IDRS.

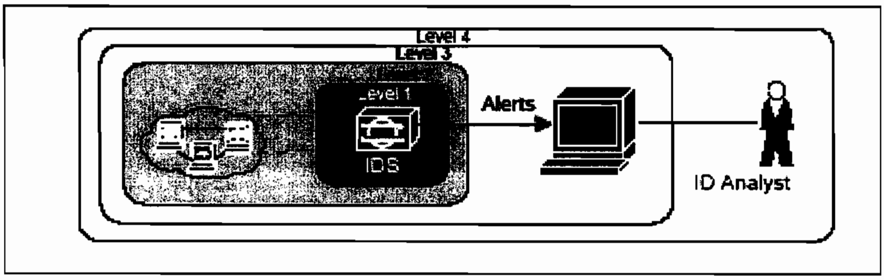

Figure 3. Evolution of the scope for addressing false positives problem.

Level 1 - Improving IDS themselves. Most of the efforts focused on building better sensors, i.e., sensors that detect more intrusions or sensors with very low false-positive rates. In contrast to building general, all-purpose IDSs, specialized IDSs focus on particular types of intrusions or attacks promising very low false-positive rates. For example in [31], they proposed a network-based IDS that exclusively focus on low-level attacks such as reconnaissance scans and denial-of-service attacks. Billygoat [32], differently, focuses on detecting worms and viruses. Whilst [29] and [33] focus on introducing better signatures. Finally, researchers in [34,35] proposed IDS targeted at detecting application-level SQL injection attacks. Note that these special-purpose IDSs need to be complemented by additional IDSs to achieve a comprehensive attack coverage, which then creates the need to deploy and maintain a heterogeneous network of complementary IDSs.

Level 2 - Leveraging the Environment. By using information about the environment (provided by vulnerability scanners, OS-fingerprinting or asset databases), IDSs can better understand the environment and significantly lower their false-positive rates. For example, work by [28] showed that without knowing how target hosts handle certain anomalies in network packets, intruders can efficiently use fragmentation to avoid being detected by network-based IDS. Addressing these concerns, Active Mapping [36] builds profiles of the environment, which can then be leveraged by IDSs. Context signatures [33], on the other hand, can understand application-level protocol interactions and thus determine the impact of an attack. A similar effect can be achieved by correlating alerts with vulnerabilities for instance in [13,37].

Level 3 - Alert Post-processing. This level uses alerts generated by IDS as input and tries to improve their quality by processing them. This includes systems using data mining and so-called ACSs. For example in the data-mining space, Julisch [8] showed how root cause analysis can be used to effectively discover large groups of false positives and remove up to 70% of reoccurring false positives in the future. Moreover, AC, in addition to false positives, addresses another problem of IDS, namely, the redundancy in alert stream. ACSs (e.g., [4,13,38,39]) aim at producing high-level alerts and thus reduce the total number of alerts the IDS generates.

Level 4 - Analyst's Involvement. Alerts generated by IDSs are passed to the human analyst to be analyzed in real time or with only a short delay. Very few related work on this level. Examples of such work are in [12,40, and 41]. Pietraszek [12] introduces a novel paradigm of using machine-learning techniques to reduce the number of false positives in ID by building classifier learning from a human analyst and assisting in the alert classification.

Building better IDS (as in *Level 1*) that has a small number of false positives is an extremely difficult task [27]. Using vulnerability scanner (as in *Level 2*) will leave the network at a high risk because it can also be used by an attacker to gather information about potential targets to identify some vulnerability that could be exploited to start an intrusion scenario [42]. While analyst's involvement (as in *Level 4*) could bring some delay (time-consuming) and unreliable as human's knowledge and experience is vary to one and another. Because of these reasons, a more practical solution to reduce the number of false positives in ID is to work on the IDS alerts, by using alert post-processing (*Level 3*) or known as analysis algorithms [43]: such as data mining and machine learning [27,44], data cleansing [45], data aggregation [38], and AC [7,39,46,47]. Thus, the main problem in this issue is *"how to filter out the false positives alerts and aggregating/grouping the redundant alerts"*. Technically, false alerts will be discarded and grouped redundant alerts are merged into a meta-alert, so that the number of alerts left to be analyzed is decreased.

3.2 Issue 2 : Constructing Attack Scenarios

In situations where there are intensive intrusions, not only will actual alerts be mixed with false alerts, but the amount of alerts will also become unmanageable. As a result, it is difficult for human users or IRSs to understand the alerts and take appropriate actions. Therefore, it is necessary to develop techniques to construct *attack scenarios* (or steps that attackers use in their attacks) from alerts and facilitate intrusion analysis [48]. In conjunction with that, the main problem is *"how to relate the alerts to one another so that they can build relationship to reveal the step-by-step attack scenarios"*. Thus, human analyst will understand what are the step by step (previous) attacks has been launched in order to make the next attack is successful. Several AC methods have been proposed to address this problem. They are:

Similarity-based approach [39,44] correlates alerts based on the similarities between alert attributes. The similarity between attributes is determined by the similarity function defined. Though they are effective for correlating some alerts (e.g., alerts with the same source and destination IP addresses), they cannot fully discover the causal relationships between related alerts.

Data mining approach [49] bases AC on attack scenarios specified by human users or learned through training datasets. Obviously, these methods are restricted to known attack scenarios. A variation in this class uses a consequence mechanism to specify what types of attacks may follow a given attack, partially addressing this problem [38].

Causal relationship approach [4,7,24,25] is based on the *cause-effect* idea. It correlates alerts if the pre-condition of some later alerts are satisfied by the consequences of some earlier alerts. This approach can potentially uncover the causal relationship between alerts, and is not restricted to known attack scenarios.

Pre-conditions and post-conditions approach is usually depends on rules or predefined knowledge. Templeton and Levitt [24] propose a *requires/provides* model (JIGSAW) for modeling chains of network exploits. The model links exploits into scenarios, with earlier exploits supplying the prerequisites for the later ones. The ***requires*** part corresponds to the action preconditions/causes, and the *provides* part corresponds to the action effects. They describe a language for specifying exploits and point out that relating seemingly innocuous system behavior to known attack scenarios can help discover new exploits. And, they do not consider combining their attack scenarios into attack graphs. However, several problems make it difficult for JIGSAW to be a practical AC technique. First, JIGSAW requires all the preconditions of an attack to be satisfied in order to consider its consequences. This is theoretically acceptable; however, it has a negative impact on AC in practice. In particular, if the IDS fails to detect one of the attacks that prepare for later attacks, JIGSAW will miss the opportunity to correlate the detected attacks.

The MIRADOR [4,25] approach also correlates alerts using partial match of *pre-conditions* and *post-conditions* of attacks, which are derived from attack database described in LAMBDA [25], and uses 'ontology rules' to represent the implication relationships between predicates. An additional notion of abductive correlation has been introduced in [4] to deal with attacks missed by IDS.

The work closest to MIRADOR is the AC method by [7,48]. They used rule-based AC which *pre-requisites* and *consequences* of the each alerts need to manually define. Intuitively, the pre-requisite of an intrusion is the necessary condition for the intrusion to be successful, while the consequence of an intrusion is the outcome of the intrusion if it is successful. Once the pre-requisite of a certain attack is detected by the correlation engine, it would be possible to predict the consequence of the attack. In the same way, it might be able to merge them and consequently detect a higher-level correlated attack. They use SQL commands for processing the rules and perform AC. Their architecture [7] is shown in Figure 4. It consists of a knowledge base, an alert preprocessor, a correlation engine, a hyper-alert correlation graph generator, and a visualization component. All these components except for the visualization component interact with a Database Management System (DBMS), which provides persistent storage for the intermediate data as well as the correlated alerts. To save development effort, they use the GraphViz package [50] as the visualization component to generate attack graphs.

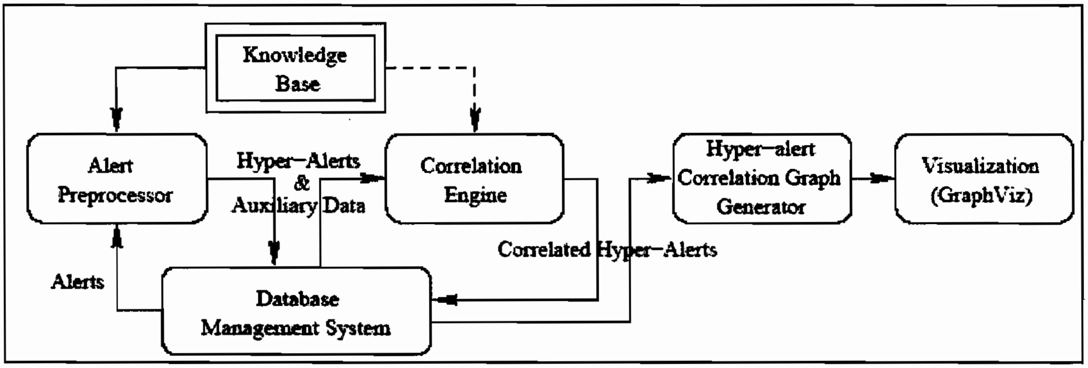

Figure 4. The architecture of hyper-alert correlation system.

3.3 Issue 3 : Extracting Attack Strategies to Predict Attacker's Next Goal

Attack scenarios that are successfully directed (or connected) can reveal the attack strategies and are normally represented as attack graphs (e.g., attack tree or directed acyclic graph). These attack graphs can be manually constructed by security experts using knowledge such as topology and vulnerabilities of the protected network. But, this approach is time-consuming, error-prone, tedious and impractical [51]. Consequently, a number of AC techniques have been introduced in order to help security analysts to learn strategies and patterns of the attackers [39,48,49,52]. However, all of these approaches have their own limitations. They either cannot reveal the causal relationship among the alerts (they simply group the alerts into scenarios), or require a larger number of predefined rules in order to correlate alerts and generate attack graphs [16]. Therefore, the problem in this issue is *"how to automatically construct attack graphs based on the correlated alerts"*.

Several research attempts on generating or constructing attack graph are by [7,16,51]. Both [7,51] used open source graph drawing software, called GraphViz [50] to visualize attack graphs. They need specific prior knowledge about alerts in order to construct the attack graph. In addition, [51] include the analysis of the attack graphs (vulnerability analysis) to validate their results. They also applied symbolic model checking algorithms [53] to automatically construct attack graph. A model of a network and its corrective properties is needed to check the algorithm efficacy.

Different from [16], they did not need any prior knowledge about alerts in order to construct the attack graph. The proposed approach is based on two different neural network approaches, namely Multilayer Perceptron (MLP) and Support Vector Machine (SVM). The probabilistic output of these two methods is used to determine with which previous alerts this current alert should be correlated. This suggests the causal relationship of two alerts, which is helpful for constructing attack scenarios. One of the distinguishing feature of the proposed technique is that an Alert Correlation Matrix (ACM) is used to store correlation strengths of any two types of alerts. ACM is updated in the training process, and the information (correlation strength) is then used for extracting high level attack strategies.

3.4 Other Issues

In this category, we listed works that are not fall into any above categories mentioned. First, work by [5] adapted main memory index structures and database query optimization techniques to facilitate timely correlation of intensive alerts. Their previous correlation system takes longer time of execution. They proposed three techniques named *hyper-alert container, two-level index,* and *sort correlation* to achieve speed correlation. Secondly, another research domain that needed in the IDRS problem space is the analysis of the attack graph. Analysis of the attack graph can be a risk analysis [78], vulnerability analysis [51,69,79]; and shortest path analysis [80]. For example in [51], they proposed two kind of analysis namely *Minimization Analysis* and *Minimum and Minimal Critical Attack Sets*. The former analysis helps analyst to determine a minimal set of atomic attacks that must be prevented to guarantee that the attcker cannot achive his/her goal. Whilst the latter analysis helps analyst to determine the likelihood that an attacker will succeed or the likelihood that the IDS will detect his/her attack activity.

4. ALERT CORRELATION TECHNIQUES

We classified the existing techniques used in AC into three categories as shown in Figure 5. AI techniques are widely and commonly explored by researchers followed by probabilistic-based technique and model checking technique. Further discussion on each technique is presented in the next subsections.

rejected regardless of the overall similarity. The new alert is correlated with the most similar meta-alert, assuming the similarity is above a specified minimum match threshold. Otherwise, the new alert starts a new meta-alert thread. Same approach used in [56], which include a statistical causality test –Granger Causality Test (GCT) to identify new alert relationship. The main advantage of using GCT is that it does not require 'a priori' knowledge about attack behaviors and how the attacks could be related. This approach can identify the correlation between two attack steps as long as the two have a high probability (not necessarily high frequency) of occurring together.

4.2 Artificial Intelligence (AI) Techniques

Refer to Figure 5, the most widely used technique in the field of AC are AI technique. AI systems or some regards is as expert systems try to reflect actions of a human expert when solving problems in a particular domain. Their knowledge base imitates knowledge of a human, which may be either 'surface' that resulting from experience, or 'deep' resulting from understanding the system behavior from its principles. Most expert systems use rule-based representation of their knowledge-base. Expert systems applied to the AC problem differ with respect to the structure of the knowledge they use.

Rule-based - Approaches that rely solely on surface knowledge are referred to as rule-based reasoning systems [57]. Ning and his researchers [48] defined a set of rules for each alerts for their pre-requisites and consequences. If any of the consequences of an alert matches the pre-requisites of next alert, then those two alerts are correlated. Similar to [38], rules describing alerts that are logically linked with each other are manually defined. Whilst in [25], they specified knowledge about attack scenarios in attack language, named LAMBDA. Other examples of such languages are STATL [58] and JIGSAW [24]. Rule-based systems, which rely solely on surface knowledge, do not require profound understanding of the underlying system architectural and operational principles. However, the downsides of rule-based systems include inability to learn from experience, inability to deal with unseen problems, and difficulty in updating the system knowledge [59].

Model-based - Model-based reasoning systems usually include some form of deep knowledge that represents relationships among system entities. As proposed in M2D2 by [46], they integrate four main types of information: information systems characteristics, vulnerabilities, security tools and events; to facilitate the aggregation and correlation process. Such approach is used as well in [60], where he extracts information from operating system-level event logs and firewall alerts together with IDS alerts for alert correlation and analysis. Their knowledge may be organized in an expandable, upgradeable and modular fashion. However, the models may be difficult to obtain and keep up-to-date.

Case-based - Case-based systems are a special class of expert systems that base their decisions on experience and past situations. They try to acquire relevant knowledge of past cases and previously used solutions to propose solutions for new problems [59]. They are well suited to learning correlation patterns [61]. When a problem is successfully solved, the solution (or its parts) may be used in dealing with subsequent problems. Pietraszek [12] used the concept of training an alert classifier using a human analyst's feedback and show how to build an efficient alert classifier using machine-learning techniques. They analyze the desired properties of such a system from the domain perspective and introduce ALAC, (an Adaptive Learner for Alert Classification), and its two modes of operation: a *recommender mode*, in which all alerts with their classification are forwarded to the analyst, and an *agent mode*, in which the system uses autonomous alert processing. However, time inefficiency (waiting for human analyst's feedback) may make them unusable in real-time alert correlation.

Neural Networks – Neural networks, which are systems composed of interconnected nodes called neurons, try to mimic operation of a human brain. They are capable of learning [62] and resilient to noise or inconsistencies in the input data. The disadvantage of neural network systems is that they require long training periods [61,63], and that their behavior outside their area of training is difficult to predict [63]. Some researchers [16] used MLP and SVM to group alerts and also to represent the correlated alerts in a way that they reflect the corresponding attack scenarios. Both MLP and SVM have their own strengths for AC. When knowledge for assigning accurate probabilities to training data is available, MLP can produce more precise correlation probabilities. Labeling training patterns for SVM is much easier but the outputs are less accurate than the ones produced by MLP. Another advantage of using SVM is that its training speed is fast and it is possible to incrementally update it in a real time environment.

Hybrid Soft-computing – This technique becomes popular nowadays in several research domain such as data mining (eg., [64]), controller design (eg., [65]) and property estimation (eg., [66]). But in the AC research field, only [67] used Neuro-Fuzzy technique to reduce false positives alerts in IDS. They showed that the techniques effectively reduce the false positives rate but not the false negatives rate. The idea of using such combination is because fuzzy sets are famous at modeling human reasoning and provide a natural mechanism for dealing with uncertainty, whilst neural networks are robust to noise and have a good ability to model highly non-linear relationships [64]. Moreover, the fusion of neural networks and fuzzy logic provides learning as well as readability [68]. On the other hand, these soft computing techniques also have some restrictions that do not allow their individual application in some cases. Fuzzy sets are dependent on expert knowledge. The training times of neural networks are excessive and tedious when the input data are large and most neural network systems lack explanation facilities.

4.3 Model Checking Techniques

Model checking is an automatic technique for verifying finite-state reactive systems, such as sequential circuit designs and communication protocols. Specifications are expressed in a propositional temporal logic, and the reactive system is modeled as a state-transition graph. An efficient search procedure is used to determine if the specifications are satisfied by the state-transition graph.

Model checker - Typically, the user provides a high level representation of the model and the correctness specification to be checked. The procedure uses an exhaustive search of the state space of the system to determine if the specification is true or not. Given sufficient resources, the procedure will either terminates with the answer *true*, indicating that the model satisfies the specifications or give a counter example execution that shows why the correctness formula is not satisfied [69]. Like [69], they used model checking-based technique to automatically construct attack graphs. Although it helps facilitate the task of defining attack graphs, it has the limitation of scalability especially for larger network and systems [70].

5. COMPARATIVE ANALYSIS ON CURRENT ALERT CORRELATION SYSTEMS

At present no standard performance evaluation strategy exists for ACS [71]. Besides, no dataset explicitly designed for testing AC algorithms is publicly available and creating such dataset in the field of analyzing intrusion alerts remains an open research problem [72]. As a consequence, a direct comparison with results in the literature is rather difficult. Researchers usually evaluate their algorithms by performing some experiments on a typical network scenario, and assessing the effectiveness of the proposed techniques on such a scenario [73]. The comparative study in this paper involved only the most representative work in AC area. Please see Table 1 to see the comparison of existing ACSs and its analysis in the next subsection.

Note that not all the researchers treat correlation issues the same way. Because of that, we compared the existing ACSs based on five important operations in an ACS and the definitions for each of them are in the following:

1) *Normalization* : a process which converting and representing alerts in a standard format. Most of the surveyed research groups are using some kind of standard format for their alert reporting. Furthermore, most groups are using some variant of Intrusion Detection Message Exchange Format (IDMEF) to suit their needs. The use of markup languages, in this case Extensible Mark-up Language (XML), simplifies the correlation process in that it is easy to identify attributes that are up for correlation.
2) *Verification* : a matching and filtering process that can remove/reduce the false positives alerts. False alerts give false correlation results.
3) *Aggregation* : a process that grouping alerts constituting a single attack and replacing them with a single meta-alert. The alerts can be from one or multiple IDSs. This steps can eliminates redundancy of alerts. Some referred it as a fusion process.
4) *Correlation* : discovers the relationship among alerts or attacks to construct the multi-stages attack scenarios.
5) *Attack Scenario Analysis* : analyzes attack scenarios resulted from prior alert correlation. For example an approach to quantitatively analyze and rank various attack paths and inform human analysts of the ones with the highest likelihood.

5.1 Problem in existing Alert Correlation Systems

As presented in Table 1, most of the proposed approaches (e.g., in [4, 7, 13, 15, 25, 38, 45, 48, 52]) have limited capabilities because they rely on various forms of predefined knowledge of attacks or attack transition patterns using attack modeling language or pre-conditions and post-conditions (some referred as prerequisites and consequences) of individual attacks. Therefore, those approaches cannot recognize a correlation when an attack is new or the relationship between attacks is new. In other words, these approaches in principle are similar to misuse detection techniques, which use the signatures of known attacks to perform pattern matching and cannot detect new attacks. It is obvious that the number of possible correlations is very large, potentially a combinatorial of the number of known and new attacks. It is infeasible to know 'a priori' and encode all possible matching conditions between attacks. In practice, the more dangerous and intelligent adversaries will always invent new attacks and novel attack sequences [74]. Therefore, a significantly better alert correlation algorithms need to be developed that can discover sophisticated and new attack sequences.

On the other hand, in the network management system (NMS), most event correlation techniques were also depend on various knowledge of underlying networks and the relationship among faults and corresponding alerts. In addition, in an NMS, event correlation focuses more on alerts resulted from network faults that often have fixed patterns. Therefore, modeling-based or rule-based techniques are mostly applied in various correlation systems. Whilst in information security, alerts are more diverse and unpredictable because the attackers are intelligent and they can use flexible strategies. Therefore, it is difficult to apply correlation techniques developed in NMS to the analysis of security alerts [74].

In fact, a correlation approach is often composed of multiple correlation approaches. There is not a unique solution that is the 'best', in terms of precision and/or complexity, to solve a generic problem of alert correlation [71]. Recent researches indicate a tendency for the adoption of combinations of different approaches for the solution of the problem in complex networks [75].

Table 1. Comparative analysis of existing ACSs.

References	Techniques	Operations					Remark
		1	2	3	4	5	
[4, 45, 25]	Rule-based	√		√	√		- using attack language called LAMBDA to specify known attack scenarios
[7, 48, 52]	Rule-based		√	√	√	√	- manually defined the prerequisites and consequences of each alerts; a massive work
[13]	Rule-based	√	√	√	√		- hard-coded rules - hard to update
[38]	Rule-based		√	√	√		- rules specified manually, very labor intensive
[15]	Rule-based			√	√		- need to maintain and update rules in database - only deal with known attack scenarios
[46]	Model-based			√	√		- integrates various types of security information - need to define prerequisites of intrusions

Table 1. Comparative analysis of existing ACSs. (cont'.)

[70]	Statistical-based	√	√	√	√		- use attack sequence pattern mining - introduce correlativity value which based on statistical equation, to relate alerts - not clear how parameter value in the equation is determined
[39]	Probabilistic-based	√			√	√	- group alerts based on similarity of attributes - thus cannot reveal causal relationship between alerts
[49]	Probabilistic-based & Case-based			√	√	√	- use various data mining techniques for learning from labeled data. The best is decision tree. - labeling alerts is manual, tedious
[56, 74, 76]	Statistical & Probability based	√	√	√	√	√	- use Bayesian and time series-based causal analysis algorithm, GCT, no predefine rules - not clear how they reduce false alerts - can discover new attack relationship

6. OPEN RESEARCH PROBLEMS

Several ACSs have being developed but there is currently no silver-bullet solution to the AC problem [77]. Based on the survey and analysis conducted, the following research directions are worth to be explored:

1) *Real-time correlation* : Most of the previous works are based on off-line correlation, which alerts are first collected and stored in a databases. It is cost-effective and resource-friendly. Although real-time or on-line correlation uses larger memory, it is worth to be explored if one wants to apply an effective real-time response plan.
2) *Adaptive correlation* : Most attacks are at the application level, and they are best detected by the host-based IDSs. It is believed that the cooperation of multiple IDSs may improve the detection rate. As a consequence, more accurate alerts shall contribute to more accurate and reliable ACS.
3) *Datasets and validation* : At present no standard performance evaluation strategy exists for ACS. Besides, no dataset explicitly designed for testing AC algorithms is publicly available. How such a dataset can be produced and validated is an open research problem as well.

7. CONCLUSION

The aim of this survey is to provide adequate information and review to newcomers to the field as well as a good reference guide for security analysts. Important concepts, challenges, existing techniques and possible research directions in the field of AC are thoroughly discussed. It is important to note that not all the researchers treat AC issues the same way. The automation of alerts analysis and correlation certainly improve the alerts quality and reveals the sequence of attacks launched by attacker. This step is very important to facilitate analysts in finding the most appropriate response and preventive mechanisms or plans such that severe damages can be minimized or prevented. This survey showed that a significantly better AC algorithms need to be developed that can discover sophisticated and new attack sequences.

8. REFERENCES

[1] P. Kabiri, and A. A. Ghorbani, "Research on Intrusion Detection and Response: A Survey", International Journal of Network Security, Vol.1, No.2, 2005, pp. 84–102.

[2] S. Manganaris, M. Christensen, D. Zerkle, and K. Hermiz, "A Data Mining Analysis of RTID Alarms", Journal of Computer Networks, 34, 2000, pp. 571–577.

[3] D. Yu, and D. Frinche, "Improving the Quality of Alerts and Predicting Intruder's Next Goal with Hidden Colored Petri-Net", Computer Networks, 51, 2007, pp. 632–654.

[4] F. Cuppens, and A. Miege, "Alert Correlation in a Cooperative Intrusion Detection Framework", Proceedings of the IEEE Symposium on Security and Privacy, 2002, pp. 202-215.

[5] D. Xu, "Correlation Analysis of Intrusion Alerts", PhD Thesis, North Carolina State University, USA, 2006.

[6] Y. Zhai, "Integrating Multiple Information Resources to Analyze Intrusion Alerts", PhD Thesis, North Carolina State University, USA, 2006.

[7] P. Ning, Y. Cui, D. Reeves, and D. Xu, "Techniques and Tools for Analyzing Intrusion Alerts", ACM Transactions on Information and System Security, 2, 2004, pp. 274-318.

[8] K. Julisch, "Using Root Cause Analysis to Handle Intrusion Detection Alarms", PhD Thesis, University of Dortmund, Germany, 2003.

[9] S. Axelsson, "The Base-rate Fallacy and Its Implications for the Difficulty of Intrusion Detection", ACM Transactions on Information and System Security, 3 (3), 2000, pp. 186–205.

[10] S. Lee, B. Chung, H. Kim, Y. Lee, C. Park, and H. Yoon, "Real-time Analysis of Intrusion Detection Alerts via Correlation", Journal of Computers and Security, 25, 2006, pp. 169-183.

[11] L. Wang, A. Liu, and S. Jajodia, "Using Attack Graphs for Correlating, Hypothesizing, and Predicting Intrusion Alerts", Computer Communications, 29(15), 2006, pp. 2917-2933.

[12] T. Pietraszek, "Alert Classification to Reduce False Positives in Intrusion Detection", PhD Thesis, Albert-Ludwigs-Universit¨at Freiburg im Breisgau, Germany, 2006.

[13] F. Valeur, G. Vigna, and C. Kruegel, "A Comprehensive Approach to Intrusion Detection Alert Correlation", IEEE Transactions on Dependable and Secure Computing, 1(3), 2004, pp. 146-169.

[14] G. Tedesco, and U. Aickelin, "Data Reduction in Intrusion Alert Correlation", WSEAS Transactions on Computers, 5(1), January 2006.

[15] P. Kabiri, and A. A. Ghorbani, "A Rule-based Temporal Alert Correlation System", International Journal of Network Security, Vol.5, No.1, 2007, pp. 66–72.

[16] B. Zhu, and A. A. Ghorbani, "Alert Correlation for Extracting Attack Strategies", International Journal of Network Security, Vol. 3, No. 2, 2005, pp. 259-270.

[17] N. Stakhanova, S. Basu, and J. Wong, "A Taxonomy of Intrusion Response Systems", Int. Journal Information and Computer Security, 1(2), 2007, pp. 169-184.

[18] B. Foo, M. W. Glause, G. M. Howard, Y. S. Wu, S. Bagchi, and E. H. Spafford, "Intrusion Response System : A Survey", Technical Report, Purdue University, 2007.

[19] G. Jakobson, and M. D. Weissman, "Alarm Correlation", IEEE Network Magazine, 1993, pp. 52–59.

[20] Y. Bouzida, "Principal Component Analysis for Intrusion Detection and Supervised Learning for New Attack Detection" PhD Thesis, Ecole Nationale Sup´erieure des T´el´ecommunications de Bretagne, 2006.

[21] J. P. Pickett, and et al., "The American Heritage Dictionary of the English Language" Boston : Houghton Mifflin Company, 2000.

[22] IDMEF, http://www.rfc-editor.org/rfc/rfc4765.txt

[23] C. Kruegel, W. Robertson, and G. Vigna, "Using Alert Verification to Identify Successful Intrusion Attempts", Practice in Information Processing and Communication (PIK), Vol.27, No.4, 2004, pp. 219-227.

[24] S. J. Templeton, and K. Levitt, "A Requires/Provides Model for Computer Attacks", Workshop on New Security Paradigms, 2000, pp. 31–38.

[25] F. Cuppens, and R. Ortalo, "Lambda: A Language to Model a Database for Detection of Attacks", 3rd Int. Symposium of Recent Advances in Intrusion Detection (RAID2000), France, 2000, pp. 197-216.

[26] E. Bloedorn, B. Hill, A. Christiansen, C. Skorupka, L. Talbot, J. and Tivel, "Data Mining for Improving Intrusion Detection", Technical Report, MITRE Corporation, 2000.

[27] T. Pietraszek, and A. Tanner, "Data Mining and Machine Learning – Towards Redusing False Positives in Intrusion Detection", Information Security Technical Report, 10 (3), 2005.

[28] T. H. Ptacek, and T. N. Newsham, "Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection", Technical Report, Secure Networks Inc., 1998.

[29] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time", Computer Networks, 31(23-24), 1999, pp. 2435-2463.

[30] S. M. Bellowin, "Packets Found on an Internet", Computer Communications Review, 23(3), 1993, pp. 26-31.

[31] R. Sekar, Y. Guang, S. Verma, and T. Shanbhag, "A High-performance Network Intrusion Detection System", ACM Conference on Computer and Communications Security, Kent Ridge Digital Labs: Singapore, 1999, pp. 8–17.

[32] J. Riordan, D. Zamboni, and Y. Duponchel, "Billy Goat, an Accurate Worm-detection System", Technical Report, IBM Zurich Research Laboratory, 2005.

[33] R. Sommer, and V. Paxson, "Enhancing Byte-level Network Intrusion Detection Signature With Context", Proceedings of the 10th ACM Conference on Computer and Communication Security, 2003, pp. 262-271.

[34] T. Pietraszek, and C. V. Berghe, "Defending Against Injection Attacks through Context-sensitive String Evaluation", Recent Advances in Intrusion Detection (RAID2005), Vol. 3858 of Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 124–145.

[35] F. Valeur, D. Mutz, and G. Vigna, "A Learning-based Approach to the Detection of SQL Attacks", Proceedings of the Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA), 2005.

[36] U. Shankar, and V. Paxson, "Active Mapping: Resisting NIDS Evasion Without Altering Traffic", Proceedings of the IEEE Symposium on Security and Privacy, California, 2001, pp. 44–62.

[37] R. Lippmann, S. Webster, and D. Stetson, "The Effect of Identifying Vulnerabilities and Patching Software on the Utility of Network Intrusion Detection", Recent Advances in Intrusion Detection (RAID2002), Vol. 2516 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 307–326.

[38] H. Debar, and A. Wespi, “Aggregation and Correlation of Intrusion–detection Alerts”, Proceedings of the 4th International Symposium on Recent Advances in Intrusion detection (RAID), 2001, pp. 87–105.

[39] A. Valdes, and K. Skinner, “Probabilistic Alert Correlation”, Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID2001), 2001, pp. 54–68.

[40] A. Siraj, and B. V. Rayford, “Alert Correlation with Abstract Incident Modeling in a Multi-Sensor Environment”, IJCSNS International Journal of Computer Science and Network Security, Vol.7 , No.8, 2007, pp. 8-19.

[41] R. Smith, N. Japkowicz, M. Dondo, and P. Mason, “Using Unsupervised Learning for Network Alert Correlation”, Advances in Artificial Intelligence, 2008, pp. 308-319.

[42] F. Cuppens, “Using an Intrusion Detection Alert Similarity Operator to Aggregate and Fuse Alerts”, Proceedings of the 4th Conference on Security and Network Architectures, 2005.

[43] D. Yu, and D. Frincke, “Improving the Quality of Alerts and Predicting Intruder’s Next Goal with Hidden Colored Petri-Net”, Computer Networks, 51, 2007, pp. 632–654.

[44] W. Lee, S. J. Stolfo, and K. W. Mok, “A Data Mining Framework for Building Intrusion Detection Models”, Proceedings of the 1999 IEEE Symposium on Security and Privacy, 1999.

[45] F. Cuppens, “Managing Alerts in a Multi-intrusion Detection Environment” 17th Annual Computer Security Applications Conference, New Orleans, 2001, pp. 22–31.

[46] B. Morin, L. Me, H. Debar, and M. Ducassé, “M2D2: A Formal Data Model for IDS Alert Correlation”, 5th International Symposium, Recent Advances in Intrusion Detection (RAID2002), Lecture Notes in Computer Science, Springer Verlag, 2002.

[47] P. A. Porras, M. W. Fong, and A. Valdes, “A Mission-impact Based Approach to INFOSEC Alarm Correlation”, Proceedings Recent Advances in Intrusion Detection, 2000, pp. 95–114.

[48] P. Ning, Y. Cui, and D. Reeves, “Analyzing Intensive Intrusion Alerts via Correlation”, Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002), LNCS, vol. 2516, 2002, pp. 74–94.

[49] O.M. Dain, and R. K. Cunningham, “Fusing a Heterogeneous Alert Stream into Scenarios”, Proceedings of the 2001 ACM Workshop on Data Mining for Security Applications, 2001, pp. 1-13.

[50] GraphViz , "An Open Source Graph Generator", 1999, http://www.research.att.com/sw/tools/graphviz

[51] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated Generation and Analysis of Attack Graph", Proceedings of the 2002 IEEE Symposium on Security and Privacy (S&P'02), 2002.

[52] P. Ning, and D. Xu, "Learning Attack Strategies from Intrusion Alerts", Proceedings of the 10th ACM Conference on Computer and Communication Security, ACM Press, 2003, pp. 200-209.

[53] E. Clarke, O. Grumberg, and D. Peled, " Model Checking", 2000, MIT Press.

[54] P. P. Bonissone, "Hybrid Soft Computing Systems: Where Are We Going?", Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000), Berlin, Germany, 2000, pp. 739-746.

[55] J. Pearl, "Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference", Morgan Kaufmann Publishers, Inc., 1988.

[56] X. Qin, and W. Lee, "Statistical Causality Analysis of INFOSEC Alert Data", Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID), 2003, pp. 73–93.

[57] M. Steinder, and A. S. Sethi, "A Survey of Fault Localization Techniques in Computer Networks", Science of Computer Programming, 53, 2004, pp. 165–194.

[58] S.T. Eckmann, G. Vigna, and R.A. Kemmerer, "Statl: An attack Language for State-based Intrusion Detection", Proceedings of the 1st ACM Workshop on Intrusion Detection Systems, Athens, 2000.

[59] L. Lewis, "A Case-based Reasoning Approach to the Resolution of Faults in Communications Networks", Integrated Network Management III, Amsterdam, 1993, pp. 671–681.

[60] Zhai, Y. (2006). Integrating Multiple Information Resources to Analyze Intrusion Alerts. PhD Thesis, North Carolina State University, USA.

[61] R. D. Gardner, and D. A. Harle, "Methods and Systems for Alarm Correlation", Proceedings of GLOBECOM, London, 1996, pp. 136–140.

[62] J. A. Meyer, "Artificial Life and the Animat Approach to Artificial Intelligence", Artificial Intelligence, 2nd edition, Handbook of Perception and Cognition. New York : Academic Press. 1996, pp. 325–354.

[63] Wu, P., Bhatnagar, R., Epshtein, L., Bhandaru, M. and Shi, Z. (1998). Alarm Correlation Engine (ACE). Proceedings Network Operation and Management Symposium, NOMS'98. New Orleans, LA. 733–742.

[64] R. Li, Y. Zhao, F. Zhang, and L. Song, "Rough Sets in Hybrid Soft Computing Systems", Advanced Data Mining and Applications, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2007, pp. 35-44.

[65] P. P. Bonissone, P.S. Khedkar, and Y. T. Chen, "Genetic Algorithms for Automated Tuning of Fuzzy Controllers", A transportation Application. Proceedings Fifth IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'96), 1996, pp. 674-680.

[66] P.P. Bonissone, and W. Cheetham, "Financial Applications of Fuzzy Case-Based Reasoning to Residential Property Valuation", Proceedings Sixth Int. Conf. On Fuzzy Systems (FUZZ-IEEE'97), 1997, pp. 37-44.

[67] R. Alshammari, S. Sonamthiang, M. Teimouri, and D. Riordan, "Using Neuro-Fuzzy Approach to Reduce False Positives Alerts", IEEE Fitth Annual Conference on Communication Networks and Services Research (CNSR07), 2007.

[68] J. Jantzen, "Neurofuzzy Modelling", Technical Report, No. 98-H-874, Department of Automation, Technical University of Denmark, 1998.

[69] O. Sheyner, "Scenario Graphs and Attack Graphs" PhD Thesis, Carnegie Mellon University, 2004.

[70] W. Li, Z. Li, L. Jie, and L. Yao, "A Novel Algorithm SF for Mining Attack Scenarios Model" IEEE International Conference on e-Business Engineering (ICEBE'06), 2006.

[71] F. Pouget, and M. Dacier, "Alert Correlation : Review of the State of the Art", Research Report RR-03-093. Institut Eurecom. Sophia Antipolis : France, 2003.

[72] U. Zurutuza, and R. Uribeetxeberria, "Intrusion Detection Alarm Correlation : A Survey", Proceedings of the IADAT International Conference on Telecommunications and Computer Networks, 2004.

[73] R. Perdisci, , G. Giacinto, and F. Roli, "Alarm Clustering for Intrusion Detection Systems in Computer Networks", Engineering Applications of Artificial Intelligence, 2006, 19, pp. 429-438.

[74] X. Qin, "A Probabilistic-based Framework for INFOSEC Alert Correlation", PhD Thesis, Georgia Institute of Technology, USA, 2005.

[75] L. Lewis, "Service level Agreements for Enterprise Networks", Artech House: Norwood, MA, 1999, pp. 158-190.

[76] X. Qin, and W. Lee, "Discovering Novel Attack Strategies from INFOSEC Alerts", Computer Security (ESORICS 2004), 9th European Symposium on Research Computer Security, Springer-Verlag LNCS 3193, 2004, pp. 439–456.

[77] K. Julisch, and M. Dacier, "Mining Intrusion Detection Alarms for Actionable Knowledge", 8th ACM International Conference on Knowledge Discovery and Data Mining, 2002, pp. 366–375.

[78] B. Schneier, "Modeling Security Threats", Dr. Dobb's Journal, 1999.

[79] S. Jha, O. Sheyner, and J.M. Wing, "Two Formal Analyses of Attack Graphs" Proceedings of the 15th IEEE Computer Security Foundations Workshop, 2002, pp. 49-63.

[80] L. Swiler, C. Phillips, D. Ellis, and S. Chakerian, "Computer-attack Graph Generation Tool", Proceedings of the DARPA Information Survivability Conference and Exposition, 2000.