Computer Communications 49 (2014) 1–17

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Review

False alarm minimization techniques in signature-based intrusion detection systems: A survey

Neminath Hubballi [a,*], Vinoth Suryanarayanan [b]

[a] Discipline of Computer Science and Engineering, Indian Institute of Technology Indore, India
[b] School of Computer Science, University of Birmingham, United Kingdom

ARTICLE INFO

Article history:
Received 23 September 2013
Received in revised form 25 April 2014
Accepted 27 April 2014
Available online 9 May 2014

Keywords:
False alarms
Correlation
Intrusion detection

ABSTRACT

A network based Intrusion Detection System (IDS) gathers and analyzes network packets and report possible low level security violations to a system administrator. In a large network setup, these low level and partial reports become unmanageable to the administrator resulting in some unattended events. Further it is known that state of the art IDS generate many false alarms. There are techniques proposed in IDS literature to minimize false alarms, many of which are widely used in practice in commercial Security Information and Event Management (SIEM) tools. In this paper, we review existing false alarm minimization techniques in signature-based Network Intrusion Detection System (NIDS). We give a taxonomy of false alarm minimization techniques in signature-based IDS and present the pros and cons of each class. We also study few of the prominent commercial SIEM tools which have implemented these techniques along with their performance. Finally, we conclude with some directions to the future research.

1. Introduction

In principle, computer systems need to be designed to prevent illegal access. However, mechanism to guard systems from illegal access is a non-trivial problem. An unauthorized mechanism designed to access system resources and/or data is called intrusion and designers are called intruders. Intruders can be classified as *Internal Intruders* and *External Intruders*. *Internal Intruders* attempt to elevate their limited privileges by abusing it. *External Intruders* attempt to gain unauthorized access to system resources from outside the target network. One of the earliest work on intrusion detection in computer networks is presented by Anderson [1]. In the seminal article, the author presents a threat model which describes internal penetrations, external penetrations and misfeasance. Further, the paper discusses a surveillance system for detecting all the three types of activities. In another major work, Denning [2] describes that users have a defined set of actions and intrusions can be detected assuming the intrusions deviate from the defined set of actions.

In recent days, computer security breach events due to intrusions are increasing. An Intrusion Detection System (IDS) monitors the system activity and reports on observation of any security violations. Traditionally there are two broad classes of IDS such as signature-based and anomaly-based. The former uses a database of known attack signatures and raises an alarm whenever network traffic matches any signature [3], whereas the later uses a model of normative system behavior and observable deviations are raised as alarms [4].

Whenever an attack is detected IDS generally raises an alarm to the system administrator. The alarm contains the information describing what attack is detected, who are the target and victims of the attack. The content associated with IDS alarms varies to a great extent depending on the nature of data (host or network) and also on the type of IDS mechanism (signature or anomaly). Signature-based IDS generates rich information along with alarm whereas anomaly IDS may just identify the connection stream which is detected as malicious.

The major concern with these systems is that, they attempt to detect suspected events which results in high false alarm rate (they account up to 99% [5–8]). Studies in [9,10] found the problem of false alarms by Snort even in the DARPA 99 dataset [11] which is generated in a controlled laboratory environment. The reason attributed for this alarming number of wrong detection is because many IDS detect too many suspicious cases. In a sense, suspected events are not necessarily intrusions to the system. An IDS with improper ruleset may miss some genuine intrusions. In the IDS literature, these cases are generally termed as false alarms. False positives and false negatives indicate whether detection is

* Corresponding author.
E-mail addresses: neminath@iiti.ac.in (N. Hubballi), vinothsuryanarayanan@gmail.com (V. Suryanarayanan).

http://dx.doi.org/10.1016/j.comcom.2014.04.012
0140-3664/

spurious or a failure respectively. In the context of this paper we define the following terminology.

- *Attack:* Any malicious attempt to exploit a vulnerability, which may or may not be successful.
- *False positive:* False positive is generated when IDS raises an alarm for an unsuccessful attack attempt.
- *False alarms:* Set of false positives.

There are various reasons for false alarm generation in IDS and some of the important ones are listed below.

- Intrusion activity sometimes deviates very slightly from the normal and some cases are difficult to differentiate.
- Often the context in which a particular event has happened decides the usefulness of the alarm generated by that event. For example, "Microsoft Distributed Transaction (MDT)" service was vulnerable to intrusion of large packets, which was generating a buffer overflow. This triggers a denial of service for the MDT service. However, this vulnerability was exploitable only in the Windows 2000 operating system which was not patched with latest patches.
- Certain actions which are normal may be malicious under different prevailing circumstances. For example, network scan is normal if done by a security administrator otherwise it is abnormal.
- Many IDS not only detect intrusions but also the number of attempts of intrusions. An attempt may not necessarily lead to a compromised system. These alarms are very likely to overwhelm the administrator.
- An alarm may represent a stage in a multistage attack which may eventually fail due to various other reasons.

In addition to the above general reasons there are many reasons attributed for false alarm generation in a signature-based IDS.

- Often it is difficult to write good quality signatures [12]. A signature should be able to detect all possible variations of a pertinent attack and do not detect all non-intrusive activity. If a signature fails to match a pertinent attack it is considered as a false negative. On the other hand, if it matches for non-intrusive behavior a false positive is generated. This misinterpretation can happen under two situations.
  - Analyzing the irrelevant portion of traffic for finding a match.
  - Analyzing the wrong application data for finding a match.
- Signature writing is highly dependent on the expert knowledge. As discovery of new flaws and vulnerabilities occur continuously, to write good signatures one needs to have complete understanding of the behavior and also sufficient data to analyze. Due to this dependency, this method is always error prone.
- In most of the cases, IDS will run with default set of signatures which are not customized to the local network. Most of the vendor supplied signature databases come with a bundle of known attack signatures. The database entries should be minimized or customized based on the target system for operational efficiency. For example, if the target network has all systems running windows operating system, then signatures written to detect a Linux specific known attack can be removed.
- Latency in deployment of newly created signatures across all the IDS running computer systems is another reason. As soon as new signatures are written they need to be deployed in the signature database. Writing a signature requires expertise in understanding the semantics of attack. Thus vendor has to update signature database regularly.

Given the voluminous number of alarms, security managers often would like to prioritize alarms based on relevance and find out those alarms which have impact on target machine and defer decision on remaining alarm analysis to a later point of time or completely ignore them. This paper is a survey of such false alarm minimization techniques in signature-based intrusion detection system.

Rest of the paper is organized as follows. In Section 2, we review other related surveys and compare our work with them clearly justifying the motivation. Section 3 presents an overview of approaches for false alarm minimization in signature-based IDS. From Sections 4–11, we discuss various techniques used for false alarm minimization. In Section 12 we discuss hybrid approach of false alarm minimization which combines the best of some of the other techniques discussed in Sections 4–11 and in Section 13 we present a summary of various commercial SIEM tools in the market showing the methods currently in use along with their performance. Future research directions are presented in Section 14 with conclusions in Sections 15.

2. Prior work

An early survey [13] gives the generic architecture of alarm handling techniques. It discusses three aspects of alarm handling namely pre-processing, alarm analysis and correlation using IDMEF message format as a standard for data collection. In pre-processing step alarms are dumped into a relational database with a schema having attributes of IDMEF format. In the alarm analysis phase repeated alarms possibly coming from different IDS are removed and in final stage correlation of alarms is done. However, the study is very elementary and does not present the state of the art completely.

Limmer and Dressler [14] describe the event correlation technique from the perspective of early warning systems. The term event is used by the authors in a generic way rather than to mean IDS alarms. They refer events as the actual happenings in the network. For example, such events can come from net-flow data, port scan, IDS alarm and others like arrival of an ICMP packet, etc. They define event correlation as a technique of aggregating security related events in a centralized location and identifying relationship between them. This survey covers correlation architectures, attack intention identification, finding the scope of the attack and method of the attack. Correlation architecture can be either a centralized or distributed architecture. Attack intention is categorized either as scan, denial of service or exploitation. Scope of the attack as targeted or non targeted attack. Correlation algorithms are classified as *1*-pass, *n*-pass algorithms depending on how many times events are read by the correlation engine. However this paper does not discuss other techniques of false alarm minimization.

Sadoddin and Ghorbani [15] have given a survey of IDS alarm correlation techniques. This survey describes alarm correlation techniques from the alarm reduction point of view. Several stages of correlation are described. First one is normalization where alarms in different formats are brought into a common format, second being the aggregation in which multiple alarms are grouped and third one is correlation phase in which different correlation algorithms are used to find the relationship between the alarms. Prominent method for correlation being the Rule based correlation.

A similar survey found in [16] also covers only the correlation techniques and not the other false alarm minimization techniques.

Zhou et al. [17] describes collaborative (distributed) attacks (port scans, denial of service, distributed denial of service) and corresponding detection techniques. Although these collaborative methods include alarm correlation techniques they do not address other aspects of false alarm minimization. The study is concentrated towards detecting attacks collectively which may otherwise have missed by a single IDS. A fully distributed detection has the limitation of not having complete information for decision making.

Salah et al. [18] study correlation as a problem for different applications like network management, network security and SCADA systems. In network management SNMP data is correlated for fault localization and it gives summarized view to the system administrator. In network security, it is used for creating a consolidated security view of system by combining information from different security systems. In SCADA systems correlation is used to identify process disturbances.

Mamory and Zhang [19] survey alarm processing techniques. They describe techniques like alarm mining and various correlation algorithms as a post processing method. It also lists several types of correlation algorithms and limitations of those algorithms. One limitation of this survey is it is not exhaustive and is limited to only alarm post processing methods.

Our contributions are summarized below:

- Surveys found in the literature are bit old and many more new techniques have been proposed later. To the best of our knowledge we review up to date state of the art related to false alarm minimization in signature-based IDS.
- Existing surveys are narrowly focused and do not adequately cover all techniques of false alarm minimization. This survey provides an extensive, focused and structured view of techniques employed for false alarm minimization within the domain of signature-based IDS.
- An analysis on prominent commercial SIEM tools which uses some of the techniques surveyed is presented.
- For each category we study a list of advantages and disadvantages.
- We provide directions for future research in this domain.

3. Overview

This section presents an overview of approaches proposed over the years for false alarm minimization in signature-based IDS. A taxonomy of several techniques is shown in Fig. 1. This diagram presents a hierarchical view of works found in literature. Several methods used for false alarm minimization in literature are studied identifying similarities and/or differences in these methods. Based on this study the methods are grouped into various categories. We have also enlisted similar survey papers in the literature and differentiated them from our presentation and coverage on several categories of false alarm minimization techniques. Table 1 presents specific differences across major surveys found in the literature marked with a response (yes/no). A snapshot of various works studied under different categories is shown in Table 2. An elaborated discussion on each of these references follow in subsequent sections.

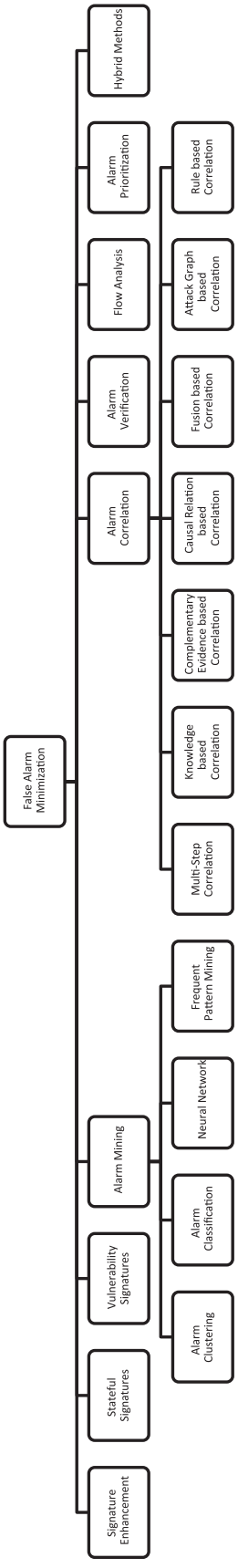

Fig. 1. Taxonomy of false alarm minimization.

4. Signature enhancement

As mentioned in previous sections, false alarms are generated due to matching of signatures in the context where it is not relevant. In order to address this, researchers have attempted to enhance the signatures with context information which can be

Table 1
Comparison with other surveys.

Technique	(Our's)	[13]	[14]	[18]	[19]	[15]
Signature enhancement	Yes	No	No	No	No	No
Stateful signatures	Yes	No	No	No	No	No
Vulnerability signatures	Yes	No	No	No	No	No
Alarm mining	Yes	No	No	No	Yes	No
Alarm correlation	Yes	Yes	Yes	Yes	Yes	Yes
Alarm verification	Yes	No	Yes	Yes	Yes	No
Flow analysis	Yes	No	Yes	No	No	No
Hybrid methods	Yes	No	No	No	No	No
Alarm prioritization	Yes	No	No	Yes	No	No

either in the form of remembering past history or network information.

In one such technique, network context information is added to the signatures, thus each signature is now stand alone and carries complete context information when the signature is valid and applicable. Sommer and Paxson [20] have developed a method of enhancing usual signatures with context information, which is termed as contextual signatures. The low level information of context is provided in the form of regular expressions. Authors argue that use of regular expressions has many advantages. Regular expressions based approach are flexible for the matching compared to the fixed strings approach. Additional information can be attached to the signature for verification since algebraic operations can be performed on the regular expressions.

A similar approach presented in [21] uses an object oriented paradigm for modeling signatures along with context information. A prototype version of the signature detection system called "passive network monitoring technique (PNMT)" is developed. Context information in the model is populated by a passive learning mechanism. For example, when PNMT discovers the target operating system of a particular host, it creates an object. Similarly, when an open port is discovered (when data is exchanged on that port) for the host in question it updates this information in the corresponding object. Signatures are written with the help of these objects by enforcing constraints on them. An object oriented paradigm based rules in PNMT is used to create new objects as and when arriving packets match the rule-set. A simple example of how these signatures differ from Snort signatures is useful for understanding the concept of context aware signatures.

Table 3
PNMT signature.

Context:	Packet
inv:	Packet.allInstances ()– >forAll (p1—p1.data.size () > 1023 and p1.tcp.destionationPort = 3372 and Session::sessionOpen (p1.ip.destinationAddress, p1.ip.sourceAddress, p1.tcp.destinationPort, p1.tcp.sourcePort) and **(IPStack::hasOS (p1.ip.destinationAddress, Windows, NT) or IPStack::hasOS (p1.ip.destinationAddress, Windows, 2000)**
implies	Alarm::logAttack (p1.ip.sourceAddress, p1.ip.destinationAddress, p1.tcp.sourcePort, p1.tcp.destinationPort, p1.time, DOS MSDTC attempt)

"Microsoft distributed transaction" service was vulnerable to large packets sent by intruders. This attack is a buffer overflow which triggers a denial of service for the Microsoft distributed transaction service. A Snort rule which is used to detect this attack is shown below.

alert tcp EXTERNAL_NET any -> HOME_NET 3372
(msg: DOS MSDTC attempt; flow: to server, established; dsize: >1023;
reference: bugtraq,4006; reference: cve,2002 – 0224; reference: nessus, 10939;
classtype: attempted – dos; sid: 1408;v rev: 10;)

Table 2
Summary of works studied.

Method	Sub category	Key references
Signature enhancement		Sommer and Paxson [20], Frederic et al. [21]
Stateful signature		Krishnamurthy and Sen [22], Vigna et al. [23], Kruegel et al. [24], Vallentin et al. [25]
Vulnerability signature		Wang et al. [26], Brumley et al. [27], Li et al. [28]
Alarm mining	Alarm clustering	Julisch [29–32], Perdisci et al. [33]
	Alarm classification	Kim et al. [34], Pietraszek and Tanner [35], Manganaris et al. [36], Pietraszek [5], Parikh and Chen [37], Benferhat et al. [38], LogRythm [39], Splunk [40]
	Neural network	Thomas and Balakrishnan [41]
	Frequent pattern mining	Sadoddin and Ghorbani [42], Soleimani and Ghorbani [43]
Alert correlation	Multi-step	Templeton and Levitt [44], Ning et al. [45–47], Cheung et al. [48], Yu and Frincke [49], Cuppens et al. [50], Cuppens and Miege [51], Mamory and Zangh [52], HP ArchSight [53], LogRythm [39], NetIQ Sentinel [54], TIBCO LogLogic [55]
	Knowledge based	Goldman et al. [56], Kruegel et al. [57], Elshoush and Osman [58], Sundaramurthy and Oh [59], Vigna and Kemmerer [60,61], Yu et al. [62], RSA Envision [63], HP ArchSight [53], LogRythm [39], McAfee Advanced Correlation Engine and Enterprise Security Manager [64], IBM Qradar [65], TIBCO LogLogic [55]
	Complementary evidence	Debar and Wespi [66], Chyssler et al. [67], RSA Envision [63], TIBCO Log–Logic [55]
	Causal relation	Qin [68], Maggi and Zanero [69], Viinikka et al. [70], Ren et al. [71]
	Fusion based	Carey et al. [72], Feng et al. [73]
	Attack graph	Ning et al. [45], Noel et al. [74,75]
	Rule based correlation	RSA Envision [63], HP ArchSight [53], LogRythm [39], McAfee Advanced Correlation Engine and Enterprise Security Manager [64], IBM Qradar [65], AlienVault Unified Security Management [76], Splunk [40], NetIQ Sentinel [54], TIBCO LogLogic [55]
Alert verification		Bolzoni et al. [77], Zhou et al. [78], Kruegel and Robertson [79], Kruegel et al. [80], Todd et al. in [81], RSA Envision [63], HP ArchSight [53], LogRythm [39]
Flow analysis		Viinikka and Debar [82], LogRythm [39], McAfee Advanced Correlation Engine and Enterprise Security Manager [64]
Alert prioritization		Porras et al. [83], Mu et al. [84], Alsubhi et al. [85], Spathoulas and Katsikas [86], Kluf [87], HP ArchSight [53], LogRythm [39], McAfee Advanced Correlation Engine and Enterprise Security Manager [64], IBM Qradar [65]
Hybrid methods		Hubballi et al. [88], Gagnon et al. [89], Abad et al. [90]

This Snort rule searches for TCP packets coming from any external network to any computer inside the network on port 3372. If one packet with these characteristics is part of an open TCP session and also if the size of this packet is bigger than 1023 bytes, then Snort sends a DOS MSDTC attempt (a denial of service attempt on the Microsoft distributed transaction service) message to the network administrator.

PNMT signature for the above Snort signature is shown in Table 3. In the PNMT signature, $p1$ represents an object of a packet. This rule describes that, an alarm is raised if the payload of the packet $p1$ is bigger than 1023 bytes and if there is an active session between the source IP address and the destination IP address on port 3372 of the packet $p1$. This part is same an in case of Snort signature. The difference between the PNMT signature and Snort signature is – PNMT rule will raise an alarm only if it is able to confirm that destination host of the attack has a Windows 2000 or Windows NT operating system. This is enforced with an additional check of operating system stack (this step is shown in bold letters in the PNMT signature in Table 3).

Some of the advantages of signature enhancement based methods are:

- Analysis of the traffic with signatures as well as network context leads to low FPs.
- Even after signature is enhanced, comparison technique (signature and traffic) remains almost similar as in ordinary signature detection systems.

Some of the disadvantages of the technique are:

- Signatures are complex and modifying them is tedious and error prone. For example, if it is discovered latter that "Microsoft distributed transaction" is vulnerable also in later versions of windows, then PNMT rule shown in Table 3. needs to be modified. The process of updating signatures selectively in a local environment needs knowledge and experience.
- Signature detection system needs to be put off when signatures are updated, so the technique leads to downtime of signature detection system.

5. Stateful signatures

Conventional Signature-based IDS analyze individual network packets and try matching against a database of signatures and trigger an alarm if a match is found. By the notion of their working they fail to detect attacks that span multiple packets. Given today's attacks are more complex it is needed to analyze more than one packet and try matching against the signature. This requires for the IDS to remember what has appeared in the previous packet and is called the state of the network. Class of IDS which work by this principle are called as stateful IDS and signatures are called as stateful signatures [91,22]. Further these signatures analyze only relevant portion of the traffic rather than entire traffic. For example if a signature is monitoring any attempted cases of logging in as root to a UNIX machine it is useful to analyze only the login session of the communication rather than the whole data exchange. Multi-step complex attacks are hence detected by only stateful IDS.

An attack language STATL is descibed in [92]. It allows to specify complex attack scenarios using a high level specification. Multi-step attacks and scenarios can be specified for creating a stateful signature. STATL being a formal model represents all the events using a state transition model. States represent different snapshots in the evolution of an attack. A stateful IDS proposed for world wide web in [23] called as WebSTAT uses STATL as the core for describing complex attack scenarios. An attempt to handle state information in high speed networks is reported in [24]. It proposed to partition the network traffic and also divide the signature set into disjoint group for the analysis. In a similar spirit a hardware based implementation of IDS which can remember the state information is discussed in [25].

Some of the advantages of stateful IDS (with stateful signatures) over stateless IDS (stateless signatures) are:

- There are attacks which spans multiple packets and sessions. These attacks cannot be detected by analyzing a single packet at a time hence stateless IDS misses these cases. However, stateful IDS detects such attacks as they contain signatures which can collect evidence across multiple packets.
- Only stateful IDS can detect multistage attacks.

The drawback of stateful signatures and stateful IDS are:

- It adds to the processing overhead. Given the pace with which networks' operating speed is increasing remembering state information is really challenging.
- Time window to be used for remembering the state information is critical for detecting attacks. An automated selection of time window is not available.

6. Vulnerability signatures

Majority of state of the art signature-based intrusion detection techniques work either by string matching or regular expression matching mechanisms. New exploits which can evade the detection by existing signature detection techniques are being used by attackers in the form of metamorphic and polymorphic attacks. Conventional signature-based IDS which operates on string matching and regular expression matching fail to detect them [27,28]. In order to detect these sophisticated attacks application semantics is used in the form of vulnerability signatures. Given a vulnerability there can be many exploits possible which can exploit it. Vulnerability signatures discussed in [26–28] can detect such attacks and improve the accuracy and detection capability of IDS by utilizing rich application semantics and protocol awareness.

Shield [26] uses the characteristics of a vulnerability, to generate a signature for all variations of exploits of that vulnerability before an exploit is seen in the wild. These signatures are installed at the network IDS and it can filter the traffic exploiting a vulnerability in a particular application of host till host installs a patch for the vulnerable application.

Brumley et al. [27] take formal approach to define vulnerability based signatures. Authors show that, semantics of a vulnerability define a language. Signatures are written either as regular expressions, Symbolic constraints or Turing machine signatures. Detection is done by finding whether the current traffic pattern is accepted by the corresponding language. Authors also analyze the complexity of such signature generation and comparison.

Vulnerability based signatures are more accurate in detecting intrusions than conventional regular expression based detection. However they tend to be more computationally expensive than the conventional signature matching. In order to address this issue Li et al. [28] provide an efficient implementation of vulnerability signature based IDS by using Single PDU Multiple Signature Matching (SPMSPM) algorithm. Here PDU stands for protocol data unit. This is a open source pilot project currently having 794 http related signatures. Its performance is shown to be 1.9 Gbps on a single CPU having 3.8 GHz processing speed and scale up to 11 Gbps on 8 core machine.

Some of the advantages of vulnerability signature-based IDS techniques are:

- Partially automate the signature generation.
- Rich semantic information reduce the number of false alarms.

Some of the limitations of these methods are the following:

- No real implementation of these methods is available in the wild yet. Although NetShield [28] is available as a free software it is still a prototype implementation which can handle limited number of protocols.

7. Alarm mining

Alarms generated by signature detection systems comprise information in terms of attributes like, IP addresses, port numbers and protocol etc. Data mining techniques use these attributes and mine a set of given alarms for summarizing them into either TPs or FPs. Characteristics learnt during the mining stage are used to classify future alarms. Although there are many ways to model prominent techniques used within the domain fall either into clustering, classification, neural network based and frequent pattern mining models. These mining techniques are discussed in the following subsections.

7.1. Alarm clustering

Clustering technique uses a set of unlabeled alarms and create a set of clusters of similar type. Later meaning is assigned to these clusters as false or true alarms. Thus all alarms in the cluster are either false or true alarms. There are different clustering algorithms used in this domain. Few of the important works are described below.

In a series of articles by Julisch [29–32] clustering algorithms are used to group IDS alarms. Julisch claims that majority of the alarms are generated by misconfigurations in the software and if the cause of these alarms can be found out then action can be taken. This cause discovery is called as root cause discovery. These articles interpret that by learning the patterns of false alarms and associated root causes future alarms can be classified as false or true. Two different data mining algorithms namely episode mining and clustering algorithms are used in the experiments. Using episode mining a set of episodes are discovered and among these episodes the ones which correspond to normal system behavior can be filtered. A modified version of classical Attribute Oriented Induction (AOI) algorithm is applied to generalize the alarms. Modifications address the issue of over generalization. Generalized alarms represent the clusters. After generating clusters by generalizing over all attributes, clusters are validated. Experiments on a series of real networks revealed a good reduction in the number of alarms. Improved versions of data mining technique to find root causes of IDS alarms is reported by Al-Mamory and Zhang [93,94].

Perdisci et al. [33] describes an alarm clustering method which can classify alarms into predefined classes. Clusters of interest are formed in the beginning by creating empty lists. Alarms are added to these clusters incrementally. Every alarm is compared to the representative of the cluster called as meta alarm using a custom distance function. Alarm is assigned to one of the cluster whose distance is minimum with the alarm. In a similar objective to identify abnormal alarms generated by IDS Law and Kwok [95], Dey [96] use K-means clustering and Incremental Stream clustering algorithms respectively.

7.2. Alarm classification

This method assumes a set of labeled alarms available for training a classification algorithm. An expert usually label the alarms either as TP or FP initially and these alarms are used for training a classification. Once classification algorithm is trained, it can be used to classify future alarms.

Kim et al. [34] presents a decision tree model for alarm classification. This includes two phase activity; first, a feature constructor which extracts features for classification; a pre-processor determines highly correlated attributes and groups the alarms based on that and second, an association rule mining which extracts interesting relationship between the attributes. The system is trained with the rules generated from the association rule mining stage and tested. As the decision tree work on a feature ranking scheme, some features on which the path to the leaf is traversed find importance over others, which is an undesirable result.

Pietraszek and Tanner [35] proposes an alarm classifier to classify alarms (as TP or FP) during run time. Proposed model uses a hybrid of two approaches – one part comprises data mining of historical alarms and passing the knowledge learnt to a human expert and the other part tunes the signature engine to reflect the learnt behavior. The scheme uses various network entities for building the classifier and these network entities are represented as a topology tree much similar to the decision tree. Alarms are clustered based on the similarity of their attributes such as IP address and port numbers after traversing the topology tree. The problem with this technique is involvement of human in the loop to tune the signature detection system which limits its practical utility.

Manganaris et al. [36] proposes an IDS alarm management technique based on data mining. Alarms from multiple sensors are collected and are evaluated against a knowledge database. This method also considers the history of sensor in the evaluation process. Knowing what sort of alarms a typical sensor generates helps in identifying true security breach incidents amongst a pool of alarms. Typically a stream of repeated alarms of a particular type from a sensor may be considered as normal than a burst of alarms which are not typically generated by that sensor. Motivation is drawn from the fact that each sensor generate alarms in a different manner, the time and day of week, kind of users they have influences the alarm generation. A profile of sensor behavior is characterized and association rule mining is used for capturing the normal alarm sequence for a particular sensor. Knowledge base of a sensor is constantly upgraded with a feedback path involving human in the loop.

Pietraszek [5] considers a supervised approach of classification of IDS alarms for false alarm minimization. The proposed technique is called ALAC representing Adaptive Learner for Alarm Classification with human in the loop. It classifies IDS alarms as true or false positive with a level of confidence and present it to the human expert. With the feedback of human expert it creates new training examples and use machine learning algorithms to learn and improve the classification performance. ALAC functions in two modes namely recommender mode where it presents alarms to administrator and in the agent mode it performs additional tasks defined for the class of alarms and confidence level of its belongingness to the class. For example it may remove certain high confidence alarms classified as false. A modified version of RIPPER algorithm is used for classification.

Parikh and Chen [37] describes a classifier ensemble method and coupled it with cost minimization strategy. Tjhai et al. [97] describes a two stage classification strategy using SOM and K-means clustering to minimize false alarms.

Benferhat et al. [38] uses a Bayesian networks model for classifying the alarms generated by intrusion detection systems. Model is capable of taking expert knowledge and increasing the accuracy of classification.

7.3. Neural network approach

Neural network is an information processing method motivated by biological systems working. A neural network takes a set of inputs and generate a set of output. The input is processed through a series of interconnected intermediate processing units between which partially processed information is exchanged and when the information passes through the last stage of processing a decision is made. Typically a neural network is configured to identify patterns or classify data after training with a set of data.

Thomas and Balakrishnan [41] proposes a neural network based alarm classification technique to detect the alarm as either TP or FP. A neural network also needs labeled alarms for training initially. In addition the nature of data given to IDS is also used to assign a final weight to the alarm.

7.4. Frequent pattern mining

Frequent pattern mining is a technique to identify frequent item sets in a given transaction database. In our case, IDS alarms are transactions and frequent alarm combinations indicate a sequence which is repeating. These repeating patterns indicate the actions an intruder has tried before penetrating into the target host.

Sadoddin and Ghorbani [42] proposes a real time alarm classification scheme based on frequent structured patterns. A component called aggregator transforms raw alarms into graphs after analyzing the connectivity relationship between them. This input is fed to a frequent mining structure which extracts frequently seen patterns in the recent past and build a tree called as frequent pattern tree. The output of this component is the correlation model called as running model. This model is dynamic and keep changing over time.

Soleimani and Ghorbani [43] perform a sequential analysis of alarms to find critical alarms. Each alarm occurrence time is used to order the alarms in a sequence. A window of size w is chosen and all the alarms falling in that window which are ordered and are termed as episodes. Each episode s is represented as a triplet s, T_s, T_e where the first part s is a sequence, T_s is the time of first alarm in the sequence and T_e is the time of last alarm in the sequence. Once the episodes are discovered a rating function is used to rate the episodes and highest rated once are chosen as critical ones. A decision tree is used to model the network parameters and is used for criticality value generation.

7.5. Advantages and disadvantages of mining techniques

Some of the advantages of alarm mining for filtering FPs are:

- Training and building a classifier is relatively easy when the labeling of alarms can be done.
- Technique can be automated without involving human in the loop.
- Keep the ability of correlating previously unseen alarms.

Some of the limitations of mining schemes are:

- Labeling of alarms may not always be possible, considering the sheer number of alarms.
- Since the underlying network context is dynamic, it may not always be possible to use mining techniques. For example, even small changes in the network configuration like patching an application and OS upgrading has an impact on the threat level of a particular attack to the target network.
- Since signature detection system can trigger thousands of alarms per day scalability is an issue.
- Alarms can have mixture of numerical, categorical, time and free text attributes. Alarm mining algorithm should work well with the mixture of attributes.
- Alarm mining techniques work offline.
- Since manual labeling is error prone predictive models should withstand some level of noise in the training data.

8. Alarm correlation

Aggregating the alarms to construct the attack scenarios is another main area of work found in the literature. These works basically group bunch of IDS alarms possibly generated in different places of network and by different IDS engines and reconstruct the attack scenarios. IDS alarms normally contain the descriptive information of the problem which is too vague to makeout some concrete inference. Correlation step helps in identifying the root cause of the problem [18].

In IDS community there is no well accepted definition for alarm correlation. The term correlation has been used to mean

- Correlation of alarms from a single IDS.
- Correlation of alarms from multiple IDS [98].
- Correlation of alarms from same type of IDS.
- Correlation of alarms from different type of IDS.
- Correlation of IDS alarms with other security components alarms (e.g. Netflow) [99].

and many more views exists. We start with defining what correlation means in our context and some other related definitions in this section.

Alert/Alarm: Alarm is a message sent by a component of an intrusion detection system about the occurrence of an event [100]. The event is typically considered as unusual or malicious.
Correlation: Correlation can be seen as black box which receives a bunch of alarms possibly generated by different IDS and other security components and generates a condensed view of alarm for system administrator.

A generic view of alarm correlation is shown in Fig. 2. Various steps involved are the following.

IDS: Alarms from various IDS engines.
Alarm Normalization: Since the alarms may come from heterogeneous IDS deployed in the network it is important for them to be in a common format. There are standards proposed in the literature for reporting an alarm which keeps alarms consistent. Internet Engineering Task Force (IETF) has proposed Intrusion Detection Message Exchange Format (IDMEF) [101] for this purpose. Currently IDMEF is the commonly used standard for alarm message exchange.
Alarm Clustering: The idea is to group same alarms together [102] possibly generated from different sensors. A time window is normally considered within which alarms are merged. To measure the similarity of alarms some common attributes are considered. Normally features like source IP address, destination IP address, source port, destination port, time, attack name, service and user are considered. Since many sensors may have captured the anomalous behavior this step plays a significant role in reducing the volume of alarms. Some literature name this step as alarm aggregation and some even call as alarm fusion.

There are many types of alarm clustering techniques based on how the feature values are matched.

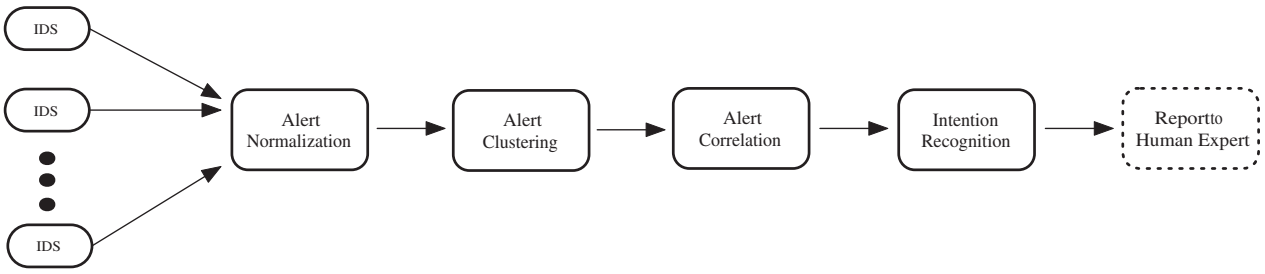

Fig. 2. Generic view of alarm correlation.

Clustering alarms caused by same event: Two or more alarms can be clustered if they belong to a same event. This may happen because more than one alarm may be generated from one event or multiple IDS sensors have access to the same event.
Clustering based on common vulnerability: If two or more alarms belong to a common vulnerability such alarms can also be grouped. One of the easiest way to find such common alarms belonging to same vulnerability is using standard CVE or Bugtraq identifiers.
Clustering based on TCP session: Alarms generated in a single TCP session can also be grouped.
Simple attribute Matching: There are two types of attribute matching. The first category called as exact matching [57] normally verifies equality condition for the feature values in order to group. In other words two alarms are similar if they have equal values for all of their features. Some exclude the time while verifying equality [57]. In the second category of attribute matching, inexact matching is done. This is because finding equal values across all the features is sometime difficult, so nearby alarms are clustered. A similarity function or distance measurement function is normally defined. For example a distance measure can be a hamming distance or edit distance.

Alarm Correlation: Analyze the clusters of alarms and provide an output by merging some clusters which are related.
Intention Recognition: Identifies the plan the attacker has.
Report: Generate a condensed view of attack scenario to the administrator.

Alarm correlation techniques can be further grouped under 6 classes. The following subsections elaborate these 6 types.

8.1. Multi-step correlation

These schemes assume that there are sequence of actions followed by the attacker before breaking into the system. The first step is a kind of prerequisite for the next step in the overall attack. Attackers actions may be unknown but the actions taken may result into alarm generation which when interpreted can assert the actions of attacker. This scheme can group complex attacks together involving a series of actions. Few research papers in the literature call Multi-step correlation as Intention Recognition, few other papers make a separation between them as Multi-step correlation is just finding the logical sequence between alarms and the goal of Intention Recognition is to find out hackers plan or intention at an early stage and report to the administrators so that further damage can be prevented. Intention Recognition help the administrator in understanding the ongoing activity and will be in a better position to defend against it. Normally Intention Recognition is a post step of alarm fusion. Although there is a thin line difference between the Multi-step correlation and Intention Recognition we study them together.

Templeton and Levitt [44] uses a *requires/provides* model for attack representation. The objective is to predict attacks. Each attack is described as a *concept* and there are certain pre-requisites called as *capabilities* that must occur before the *concept* is enabled. *Capabilities* are the generalized templates rather than individual attack models. They accumulate the symptoms of the attack rather than the attack itself. *Concept* is a subtask in an attack and there are requirements that must be satisfied for an attack to be detected. Thus the *requires/provides* model can predict all variants of the attacks as long as the goal of the sequence of events is same. Another article by Reynolds et al. [103] uses same *requires/provides* model and provide a model for generation of predicates. Similar models are presented in [45–47] where Ning et al. also follow the same assumption which is – "In a series of attacks earlier attacks gives rise to later ones". This relationship is captured in the form of *pre-requisites* and *consequence* model using predicates.

In [48] an attack modeling language is presented. This model describe two constructs called as precondition and post condition. Two alarms are correlated if the post condition of alarm one is matching with the pre condition of another alarm. With this approach a logical relation can be established between alarms. One of the main limitations of this approach is, it requires manual writing of preconditions and postconditions for all possible alarms that can be generated. This can be sometimes erroneous. In a similar approach [49] modeled these scenarios with extended Hidden Colored Petri-Net (HCPN). HCPN model has 11 tuple and uses the observation probability of a particular alarm rather than just a plain set of precondition and post conditions.

Cuppens et al. [50] model attacks as a set of transitions in order to achieve a goal known as intrusion objective. Since the transition is systematic whole set of actions can be written in the form of logic equations. Similar to [48] this method also uses precondition and postcondition to correlate the alarms.

Cuppens and Miege [51] proposes a multi-step correlation algorithm called as CRIM. CRIM models the attacks using modeling language LAMDA [104]. Similar to [48] this method also uses precondition and postcondition to relate the alarms. Two different approaches are used for correlation based on the level of information available with alarms. In the explicit correlation system administrator is able to establish some relation between the events and in the implicit correlation analysis of events reveals some connection between the alarms. A similar approach by Mamory and Zhang [52] describes a modified version of LR parser to build the attack trees representing attack scenarios.

Few advantages of this group are

- It can recognize complex attack scenarios which cannot be detected by a single rule or signature.
- It reduces false alarms reported to the administrator.
- It is a systematic approach and less error prone if properly defined.

Few drawbacks of this approach are

- Since these methods group alarms based on sequence ordering a missing precondition or post condition may result into a uncorrelated alarm. A precondition or post condition may be missed due to false negative generated by an IDS. Formal models proposed do not discuss how such cases can be handled.
- Manually creating production rules to determine the precondition and postcondition for every alarm is not possible.

8.2. Knowledge-based correlation

These schemes use an extensive knowledge base of systems being monitored. This knowledge base can be either static or dynamic. Knowledge base is in the form of target systems operating system, kind of applications running and some time their known vulnerabilities (some literature refer this information as meta data [14]). Knowledge based correlation schemes group the alarms which are similar based on some similarity metric [105]. A set of identified features are used for the distance measurement. In the first phase low level events generated by IDS are clustered based on the similarity measure, then over all scenario of the attack is visualized. This initial phase is called as thread reconstruction. The idea behind this thread reconstruction is to fuse the detection of an ongoing attack by multiple sensors together. By fusing multiple alarms of the same type into *hyper alarm* or *fusion alarm* security administrator is left with a minimal information yet pointing to the issue.

In [56] a method to capture all the security policies and alarms generated by multiple IDS is proposed. An architecture called as *SCYLLARUS* is discussed which uses a component called as Intrusion Reference Model (IRM). IRM has static and dynamic information such as event database, configuration of the hardware and software of the site that is being monitored and also a security goal database. First component in IRM known as Cluster Preprocessor assembles a set of related IDS alarms. The next component known as Accessor will examine the set of events from events database and finds relation between the alarms and events to determine plausible alarms. Once a set of plausible alarms is determined, impact of these alarms on the target network is assessed by evaluating these alarms against policy goals of the target network.

Sourcefire the company owned by Snort's author has an intrusion prevention product called as Sourcefire IPS [106] which also passively learns the network assets and automatically tunes the IDS. An intelligent engine can automatically pick and enable required Snort rules based on the network assets.

Kruegel et al. [57] have reported the notion of alarm correlation.[1] Focus of the work is to provide a model for alarm analysis and generation of a prioritized report for system administrator after analyzing the impact of underlying attacks on the target network. This work is more comprehensive and involves steps such as

- Normalization: Brings the IDS alarms into a common format.
- Preprocessing: Checks the values assigned to various attributes are consistent.
- Alert fusion: Groups the IDS alarms based on similar attributes.
- Alert verification: Verifies the success of the alarm in the network contest.
- Thread reconstruction: Group related alarms into one thread.
- Impact analysis: Scores the alarms based on what damage it can cause to the attacked resource.
- Attack prioritization: Prioritize the reconstructed alarms.
- Report generation: Generates the report for the system analysis for viewing.

Elshoush and Osman [58] and Sundaramurthy and Ou [59] also describe similar approaches for correlation.

NetSTAT [60,61] is a machine which models the network attacks as state transitions. It also models network environment and systems to be protected as hypegraph. Since the network state and attack scenarios are formally described it helps in understanding what events need to be monitored for the attacks to be detected. NetSTAT includes the three components as *Network Fact Base* – which contains the network topology information, i.e. hosts protocols and their services, etc. *State Transition Database* – Includes scenarios to be detected as attacks. *Probes* – are general purpose IDS which monitors the network traffic and reports to the administrator.

A formal model based on logical relationship between hosts, events, vulnerabilities and other complementary events is described in [107]. It provides a framework for correlating the vulnerability scanner output, CVE and Bugtraq information too with NIDS alarms. On the same lines Yu et al. [62] also uses a precomputed knowledge base of target network to correlate alarms.

8.3. Complementary evidence based correlation

Complementary evidence based correlation techniques correlate the IDS alarms with other type of security components evidence. This also involves correlation between IDS which work under different principles; for example correlating alarms of NIDS with HIDS and even proxy server and host syslog information, etc.

In [66] a detailed model for correlation of IDS alarms specifically for two commercially available IDS is discussed. The overall operation is divided into layers. First layer is known as probe layer which is an IDS deployment place. Next level layers are alarm correlation layers and these correlate alarms from various probe layers. Entire correlation process is organized as a tree. Alarm correlation layer implements correlation algorithms where the correlation criterion is defined in terms of rules. A separate and detailed object oriented model is given for collecting information from various units such as source of the attack, destination of the attack and whether attacker is targeting only single host or multiple hosts. Aggregation layer has aggregation rules and generate aggregated view of severity of the events happening in the network.

Chyssler et al. [67] have used a framework for correlating syslog information with alarms generated by Host based Intrusion Detection System (HIDS) and Network Intrusion Detection System (NIDS). Correlation process begins by removing alarms through a filtering process which are known to be non effective. Following this, both HIDS and NIDS events are matched to determine success of the attack attempt.

University of Illinois studied alarms generated by various security systems including BRO IDS at the National Center for Scientific Agency [108]. A summary information of the trends observed on credential stealing attacks was published. This work used alarms from security components like BRO, netflow, syslog data and file integrity monitoring for the study. Incidents which are nearby in time are correlated to build a case for a genuine security violation [109]. For example an IDS alarm with suspicious download is correlated with an incident from Netflow which shows connection with machines in the vicinity of download. This study makes an

[1] Although it includes some other techniques also for false alarm minimization we study it here as it uses a knowledge base to perform the analysis.

observation that attacks on authentication violations are the predominant attack types.

CISCO has a product in the name of Firewall Intrusion Detection [110] which integrates limited IDS functionalities with firewall to detect some attacks. These functionalities are defined in the form of rules. There are rules which detect DNS, HTTP, IP, fragmented attacks, etc.

8.4. Causal relation based correlation

The objective of causal relationship discovery is to test and identify causal relationships among variables under study. Given two or more random variables it identifies how the variables are related to each other. Bayesian network is a form of causal relationship description method which is usually shown as a Directed Acyclic Graph (DAC). Each node in the graph represent a variable and an edge between any two nodes describe the dependency that exists between the variables. In our case nodes represent alarms and edges represent relationships. By analyzing a set of alarms this correlation creates a DAC graph without any other input.

In order to address the issue of correlating unknown alarms and recognize novel attack scenarios Qin [68] proposes to integrate knowledge base correlation with statistical and temporal correlation techniques. This class of correlation focuses on discovering novel attack strategies via analysis of security alarms. In order to find causal relationship between alarms Qin describes three types of hypothesis testing. First hypothesis testing discover relationship between domain knowledge and IDS alarms, second and third hypothesis discover statistical and temporal relationships between IDS alarms.

In [111,68] a statistical model based on Bayesian Network is used to model the causality relationship that exists between alarms. IDS alarms and their relationships are represented by a graph with nodes representing alarms and causality relationships represented by edges. Hyper-alerts created after clustering identical alarms are divided into multiple alarms with a time window. Each alarm now indicates how many times a particular alarm occurred in a time slot. Edges of the graph are established after studying the causal relationship that may exist between two arbitrary alarms in a particular time slot. Causal relationship is identified by the mutual information between the alarms.

Maggi and Zanero [69] model alarms as random events in time. The objective is to correlate host based IDS alarms with NIDS alarms. Two events which are near in time are correlated subject to two statistical hypothesis tests. First test constraints that, within a random amount of time, the occurring of a host alarm, it is preceded by a network alarm. If this happens they are said to be correlated otherwise they are unrelated. Second test refuses to correlate the host based alarm with the alarms of network based IDS generated against other hosts.

Viinikka et al. [70] present the assessment of regularities (in terms of stream and frequency) between normal system behavior and alarms generated out of this behavior. This regularity is modeled using an autoregressive Kalman filter. This non stationary model uses last n sample measurements to predict the current measurement. This moving history time series modeling essentially allows the model to adapt to changes in the normal behavior.

In [71] a fully automated and real time analysis setup is described. In the first step the alarm correlator is trained with a set of alarms. During this stage a Baeysian probability analysis based causality relationship is established between the set of alarms. These identified causality relationships are used to classify future alarms on the fly. In addition this method can also indicate changes in the temporal relationships and patterns that exist between the alarms. If there is a sudden surge in the number of alarms or lot of uncorrelated alarms in comparison to previously seen relationship a new set of rules can be learnt.

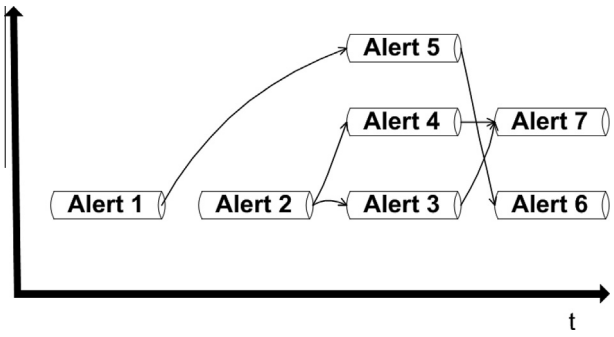

Fig. 3. Generic view of graph ordering.

8.5. Fusion based correlation

In this category of correlation, multiple layers of correlation is performed. Each correlation is based on different criteria. After the initial stages of normalization and prepossessing, a spatial fusion is done which fuses alarms from different anomaly detection systems. In the subsequent stage temporal fusion is done which orders the alarms according to their timing information. These two fusion stages reveal much more information than individual alarms.

A simple prototype implementation of alarm correlation technique involving two commodity IDS techniques can be found in [72]. It uses a database to store the alarms generated by IDS and performs tests on the alarm database. SQL queries are written to perform induction on the alarm dataset which will uncover not only the multi-step alarms but can also interpret the missing steps and alarms.

In [73] Feng et al., describes an event driven system for alarm fusion. The IDS in this case is a linguistic Subject-Verb-Object (SVO) model. Here Subject refers to the origin of an event, Verb refers to the event that the subject has performed and Object refers to the other entity involved with the Subject in doing the action. Alarms are gathered from these IDS and the following steps are performed.

(i) Preprocessing: Collect the parameters of IDS alarm in terms of SVO.
(ii) Normalization: All alarms are mapped to preselected list of alarm class and each class has a score associated with it.
(iii) Special alarm fusion: Group the alarms based on some common features and these derive a threat score.

8.6. Attack graphs based correlation

Attack graph based correlation techniques rely on the fact that vulnerability in a host when studied in isolation can reveal little information. For example if a particular host has a known low impact vulnerability many of the previous methods would rate the corresponding alarm as a low priority alarm. On the other hand hackers work through a strategic plan where in they may first break into a target host and use it as stepping stone to reach most critical systems and servers in network. This class of work identifies a set of possible paths that an attacker can take to penetrate and affect critical systems [112]. These paths are represented in the form of a graph hence are called as attack graphs. If the IDS are deployed to detect any misbehavior covering all those paths then more likely the administrator will know about the incident. In short, attack graphs show the dependencies that exists for penetrating a host and also interconnections between hosts in a network. A generic

view of attack graphs is shown in Fig. 3. The x axis indicates the time and connections between the alarms indicate the relation. There can be multiple paths between the first and last alarm in a scenario. For example from the figure we can see after alarm 2, either alarm 3 or alarm 4 is generated. In either case next alarm is alarm 7 showing more than one path for penetration.

In [45] correlated alarms called as hyper alerts are represented as a graph where the nodes represent the alarms and edges represent relation between them. Essentially the consequence of the alarm a_1 matches with prerequisite of the alarm a_2. In [113] a hybrid approach of combining multi-step correlation with knowledge based correlation techniques is proposed. Alarms may not correlate well into a correlation graph if the prerequisite and consequence of alarms do not match properly, this may lead to them being distributed across multiple correlation graphs. In [113] this is handled by merging different correlation graphs based on the inputs of knowledgeable correlation steps. In [114], paths in the graph are searched to identify different attack scenarios.

Noel and Jajodia [74,75] proposes to learn the topology of network, operating system and other configuration information and predict the possible paths which an attacker can use to penetrate the system. The alarms generated from IDS can be correlated with known vulnerability paths to decide whether a serious security breach has happened. These attack graphs also helps in identifying key places where detection sensors need to be placed and also identifying where security hardening need to be done. In [115] a formal framework is described to construct attack graphs and design countermeasures for hardening the network security.

8.7. Rule-based correlation

In this correlation method experts rules are encoded as if–then–else statements. These rules are similar to multi-step correlation rules. Unlike multistage correlation, this method relaxes the sequence ordering of events. However these kind of rules are useful where configuration of system are not altered frequently. For example a rule can be written to count the number of events by a firewall.

Rule Type: Repeat Attack-Firewall
Goal: Early warning of scan, worm propagation, etc.
Trigger: Alert on 15 or more Firewall Drop/Reject/Deny Events from a single IP Address in one min.
Event Sources: Firewalls, Routers and Switches

In the above rule, one event does not become the base for next event instead after the number of events from firewall exceeds the threshold set an alarm is triggered. It is worth noticing that, many of the commercial SIEM tools heavily rely on these type of rules.

8.8. Advantages and disadvantages of correlation techniques

Advantages of correlation steps are

- Multiple related alarms for a single attack attempt are grouped. Almost all alarm correlation techniques group repeated alarms of same attack since repeated alarm messages do not add anything extra to the knowledge of the administrator.
- Present a condensed view of attack activity in the network to the system administrator.
- Reducing false positives.
- Alarms when correlated into groups can reveal interesting cases which individual alarms fail to capture.

Drawbacks of above correlation techniques are

- In multi-step correlation all the possible relationships between all types of alarms need to be encoded. This may be erroneous and some times tedious to do.
- Detailed knowledge base of the target network need to be kept updated all the time in knowledge base correlation techniques.
- In complementary evidence technique often it is difficult to establish the relationship between other security components reports and IDS alarms.
- All these methods require prior knowledge in terms of either attack scenarios, their effect on the target network or complementary evidence hence fail to correlate alarms whose relations are not established apriori.
- Correlating alarms from IDS sensors deployed at different places is often difficult. For example correlating NIDS and HIDS is often difficult because response time of NIDS and HIDS are different. HIDS will not report the intrusive activity in real time. Thus the time window used for searching a correlation event becomes an important factor. If the window is too small then some events may not correlate and if otherwise two or more unrelated events may be correlated.
- Rule based correlation is dependent on expert knowledge to write rules. Similar to multi-step correlation writing rules are difficult and error prone.

9. Alarm verification

Alarm verification is a technique where IDS generated alarm is verified to determine whether the attempted attack is successful and also does the underlying attack has an impact on target network using some verification mechanism. There can be two kinds of verification as active verification and passive verification. In the active verification as and when an alarm is generated it is verified online and in the passive verification a database of possible success cases are stored and the alarms are verified against the database.

One such proposal is given in [77]. This is an engine to correlate IDS alarms (generated by analyzing incoming network traffic) with an Output Anomaly Detector (OAD). OAD is an anomaly detector in the reverse channel. The scheme is based on the assumption that there should be an anomalous behavior seen in the reverse channel in vicinity of alarm generation time in the incoming channel. Thus, if these two are correlated a good guess about the attack corresponding to alarm can be made. The time window used to look for the correlation is very critical for correctness of the scheme; a very small time window may lead to missing of attacks while a large time window may result in increase of false positives. Further, in some scenarios no anomalous behavior is seen in the reverse channel, for example, no response from the victim.

Zhou et al. [78] uses protocol analysis on the reverse channel to verify the success of an attack. The assumption is attacks often change the behavior of target programs and results in violation of standard protocol behavior. For example a buffer overflow against a FTP server may result in yielding a shell to attacker on the victim machine. Thus the resultant communication no longer falls within the realm of accepted FTP protocol behavior. The key idea here is to use this kind of changes and decide the success of attack.

Kruegel and Robertson [79] and Kruegel et al. [80] describes an extension to Snort alarm processing unit to verify the alarms generated. Nessus vulnerability scanner is used to initiate the verification mechanism. When an alarm is generated its CVE number is used to invoke appropriate NASL script in Nessus.

Advantages of verification techniques are

- Near real time verification of alarms about their impact and usefulness on target network.
- Had high success rate of minimizing false alarms.

Drawbacks of verification techniques are

- An evasion technique with forged server responses is proposed by Todd et al. in [81]. This technique generates spurious responses so that it will interfere with verification mechanisms and makes the IDS to believe that attack has failed. This evasion mechanism is a form of mimicry attack.
- Mimicry class of attacks are possible since verification mechanisms rely on server responses. If there is no response then often it is an indication of success of attack as it will divert the execution of target server elsewhere and prevents it from responding. If the attack can send a fake response as in case of normal scenario the IDS detects no abnormal behavior and hence ignores the alarm.

10. Flow analysis

The idea of flow analysis is to analyze a set of alarms generated under normal operating environments and also under abnormal scenarios. This is due the fact that some IDS generates alarms even under normal operations. For example an ICMP packet may trigger a Snort rule and hence generate an alarm. However the number of such alarms generated will decide whether it requires an immediate attention of administrator. For example if the alarm is generated repeatedly it is likely that it needs an immediate attention rather than some other alarm which has no other information about its success. Repeated alarms indicate some pattern. By aggregating alarms which are not symptoms of attacks and appear in high volume a condensed view can be given to the administrator.

Viinikka and Debar [82] model the IDS alarms as flows and use the extended weighted moving average (EWMA) to identify bursts of alarms among the set of flows of alarms. At different instances of time τ_i alarm flows are accessed and any significant deviation in the flow with respect to the previous cases are flagged as abnormal.

An advantage of this category is

- It considers history of IDS in generating a particular type of alarm.

A disadvantage of this category is

- An attacker can make system learn the behavior by sending spurious packets to raise alarms similar to attack scenario.

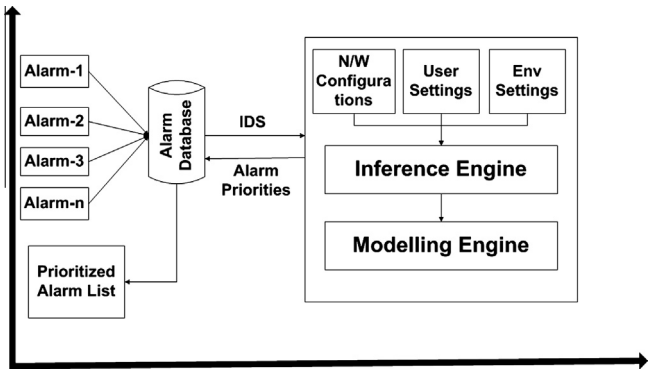

Fig. 4. An overview of alarm proritization mechanism.

11. Alarm prioritization

Prioritization techniques rate the alarms based on post assessment or some evaluation. Given a set of alarms a list containing prioritized versions is given to the system administrator. Different aspects are considered for generating this rate value. For example target network topology, IDS history, placement of an IDS in the network, etc. are generally considered. Fig. 4. shows a conceptual view of prioritization technique. It has an alarm database and various configuration and settings as input. Inference and modeling engines represent this information in the format suitable for prioritization. Output of this step is a list of prioritized alarms.

Porras et al. [83] proposes an alarm ranking technique known as M-Correlator. This scheme considers three types of information namely, (i) alarms from different security systems like signature detection systems and firewalls, (ii) network configuration (in terns of vulnerabilities vis-a-vis IP addresses, port numbers, application/OS in execution, etc.) and (iii) user defined parameters like criticality of applications, amount of interest in a type of attack. All this information is correlated to generate the rank of a particular alarm. Different lists of alarms prioritized by various parameters are generated. Also, related alarms can be grouped together using a clustering algorithm to generate a consolidated list of alarm messages.

Mu et al. [84] presents a similar topology tree to represent the network information. Authors claim that correlating topology information with alarm is often incomplete due to various reasons. If incomplete information is used to correlate alarms with topology information then this may induce errors in the final decisions. To avoid that a matrix computed by correlating attributes of alarms and topology information is created. With some mathematical operations a relevance score is derived with the help of this matrix. In an another work by the same authors [116] proposed to use Dempster–Shafer theory to handle the uncertainty in the correlation.

Alsubhi et al. [85] proposes a fuzzy logic based alarm prioritization technique. This technique also generate a priority score by considering various metrics into account. Namely the following metrics are considered for score generation – (1) Applicability – deals with whether a particular attack is relevant in the context; (2) importance – deals with how important the target entity is; (3) status of sensor – takes into account whether IDS from which the alarm is generated is well configured or not; (4) severity – how severe the attack is and also (5) historic view of IDS. The advantage of this method is it can work even in the environment where imprecise data is available because it is using many parameters into account.

Spathoulas and Katsikas [86] proposes a filter which has three components – (1) Neighboring Related Alerts (NRA), (2) High Alert Frequency (HAF) component and (3) Usual False Positives (UFP) component. The set of alarms produced by a typical IDS is fed to every component. Each component generates a score for every alarm generated by IDS indicating the probability that alarm being PF and finally all these scores are combined for final score based on which an alarm is decided as either TP or PF.

SRI International has an issued US Patent – [117] for alarm prioritization. Like all previous cases this method also considers various aspects for generating a relevant score.

A technique for ranking alarms based on voting is described in the master's thesis [87]. It considers the votes given by an administrator for a particular type of alarm and generate a rank. In order to account for balancing historic and recent votes exponential moving average technique is used while generating the rank.

Advantages of alarm prioritization are

- Substantially reducing the number of alarms which needs administrators' attention.
- Can take various aspects while assigning the rank to the alarms.

Few of limitations of alarm prioritization are

- There is no unique rating technique which is useful in all cases.
- Tuning the rating function is required for different environments.

12. Hybrid approach

False alarm reduction is essentially a subset selection problem. Given a set of alarms by IDS, the minimization scheme should automatically select a subset of these which are applicable and has impact in the target network. Such alarms we name as effective alarms and the rest are called as ineffective alarms. Since the network infrastructure is dynamic in nature the set of effective and ineffective alarms and hence attacks keep changing. From the above discussion it is evident that, methods proposed in the literature do not perform well in a dynamic network environment. Data mining techniques are not always applicable to filter the false alarms.

Thus there is a need for a hybrid approach which can mix the best of filtering based schemes and data mining schemes to reduce the false alarms.

In [88], a method to reduce the false alarms without manipulating the default signature set (i.e., neither altering the signatures nor turning them off) is proposed. This method is a hybrid approach involving both filtering and classification and thus combines the strengths of both.

The network context information is decoupled from IDS by adding a layer of filtering to the alarms generated by IDS engines like Snort [118], BrO [12]. The alarm minimization with this filter is local to the network. The false alarm filter involves the following:

- A dynamic threat profile representing the vulnerabilities present in the network. This threat profile is built periodically to get the most consistent threat view of the local network. The profile can be generated by maintaining a database of all known vulnerabilities which are published in the wild. The Bugtraq and CVE databases are the best sources for this. To build a threat profile for the target network vulnerability scanners [119] like Nessus [120], GFI Languard [121], Nmap [122], and Retina [123] are used.
- A neural network based correlation engine for filtering alarms generated by IDS. This engine correlates the threat profile with alarms and filters false alarms. Initially a set of alarms are generated by IDS with some sample attack programs and these alarms are labeled. Such labeled alarms are used for training a neural network classifier.

Gagnon et al. [89] proposes a similar alarm classification technique using network context information and data mining algorithms. The target networks context information is derived from various open source tools available in the wild.

Juniper Network has products with the IDP series [124] which falls under this category. These are advanced switches which combine detection and prevention technologies together. Detection is a combination of many methods and it is called as multi modal detection. Signature-based, anomaly-based traffic anomaly-based and protocol anomaly detection techniques are combined with application awareness and sites policy enforcement. This heterogeneous combination can minimize the false alarm rate considerably.

Abad et al. [90] proposes a visualization tool which correlates a host based activity and log information to that of network view. Two tools namely NVisionIP [125] (a host monitoring tool), and VisFlowConnect [126] are used to show how the log information at different levels can be correlated to detect intrusions.

Some of the advantages of this category are

- Combine strengths of both mining approaches and other correlation techniques.
- Can thwart attacks at early stage as few of these include IPS capabilities too.

A few drawbacks of these techniques with current state of the art are

- There are different methods to capture and store configuration information resulting in a lack of standard technique.
- There are many data mining algorithms available which one in suitable for network configuration based correlation is not studied with a comparison.

13. Commercial tools

Tool	Source of data	False alarm minimization method	EPS/MPS
RSA Envision	IDS Alarms, Vulnerability Scanners, Configuration Database, System Logs	Knowledge Based Correlation, Complementary Evidence Based Correlation, Alarm Verification	30,000 EPS [127]
HP ArcSight	IDS Alarms, IPS Alarms, System Logs, Firewall Logs, Configuration Management Data	Rule based Correlation, Knowledge based Correlation, Multi-step Correlation, Alarm Verification, Alarm Prioritization	12,500 EPS [128]
LogRhythm	Network log and audit data, host activity via custom LogRhythm System Monitor, Custom SmartFlow Data, File Integrity Monitoring, Vulnerability Database and Network Forensics with Application ID and Full Packet Capture	Rule based Correlation, Multi-step Correlation, Knowledge based Correlation, Flow Analysis, Alarm Mining with AI engine and Pattern Recognition, Alarm Verification and Alarm Prioritization	75,000 MPS [129]

(*continued on next page*)

Tool	Source of data	False alarm minimization method	EPS/MPS
McAfee Advanced Correlation Engine and Enterprise Security Manager	Application Activity, Vulnerability Scanner Data, System Properties, IDS Alarms, Firewall Alarms	Flow Analysis, Alarm Prioritization via Scores, Rule based correlation, Knowledge-based Correlation	Not available
IBM-Q1/QRadar	Flow data, IDS Logs, Application Data Activity, Vulnerability Scanners, Configuration Data	Rule based Correlation, Alarm Prioritization, Knowledge based Correlation,	20,000 EPS [130]
AlienVault Unified Security Management	File integrity monitoring, Vulnerability Assessment Data, Host-based and Network-based IDS Logs	Rule based Correlation	10,000 EPS [76]
Splunk Enterprise	Firewall logs, IDS logs, OS logs, LDAP/AD, DNS logs, NetFlow Data and Email/ Web Servers Logs.	Rule based Correlation, Mining of Events, Statistical Correlation	20,000 EPS [131]
Tenable's security Log Correlation Engine	IDS Alarms, Vulnerability Scanners, Configuration Data, Web Server Logs, Firewalls, Data loss prevention solutions, Raw network traffic, Application logs, File Integrity Logs and User activity Logs	Rule based Correlation	30,000 EPS [132]
TIBCO Log–Logic	Database Logs, Application logs, Web Server Logs, Hypervisor Logs, IDS Logs, Firewall Logs	Rule based Correlation, Multi-step Correlation, Knowldge based Correlation, Complementary Evidence based Correlation	5000 EPS [55]
NetIQ Sentinel	IDS Logs, Firewall Logs, VPN Logs, Router/Switch, System Logs and Configuration database	Rule based Correlation, Multi-step Correlation	16,000 EPS [54]
Tripwire LogCenter	File Integrity Checker, Vulnerability Database, Netflow, IDS alarms, Firewall logs	Rule based Correlation, Complementary Evidence based Correlation	10,000 EPS [133]

Organizations IT infrastructure generate large quantity of logs every day and these logs may be generated by diverse security equipment. The SIEM tools collect large quantity of logs through various sensors and log them in database. The sensors can be either a custom sensor or can be collected directly through the equipment as it is. SIEM tools help in discovering user behaviors, network anomalies, system downtime, policy violations, internal threats, regulatory compliance, etc. These tools analyze event logs and syslogs automatically through the log analyzer which speed up the otherwise time-consuming and painful manual analysis.

SIEM tools are multipurpose tools which does collecting of event logs, archive them for later reference, does correlation with rules (and other methods), custom reporting and even provide methods for efficiently index the events for quick search. Some of the tools have even adapted Big Data analysis techniques like Map-Reduce [134] and Hadoop [135] for handling massive scale data. Almost all of the SIEM tools do the prepossessing and other generic steps of Correlation mentioned in the beginning of Section 8 like Normalization, Clustering of alerts[2] and Reporting.

We compare several commercial SIEM tools in terms of their feature and performance. Features include the appliance or tool's capability to collect and understand logs from different sources. We also enlist the type of technique used in their correlation Engine. We also report the performance of tool/appliance in terms of its ability to correlate number of events per second.

SIEM tools are of great help to manage and analyze events of any organization.

- Single window view: They provide a single window mechanism to correlate events from diverse sources.
- Application awareness: These tools help to quickly identify application issues attracting attacks.
- User awareness: Help identify users who are affected by the threats detected.
- Scalability: Some of the tools are capable of collecting events from hundreds of network equipment.
- Diversity: Have the capability to interface with several types of sources for data collection.
- Automated reporting: Can generate reports for applications and users.

Some of the challenges of the current SIEM tolls are following:

- Deployment: Dealing with the complexity of the product is an issue many customers are facing. It takes quite a lot of time in deploying and configuring these systems.
- Cost: Some these tools are quite expensive. Customers with low budget may not be able to afford having a SIEM configured for their networks.
- Trained manpower: Since each SIEM tool has its own interface and a method for writing rules, trained manpower is required.

[2] This is a basic level of Grouping of alarms based on attributes and not to be equated to Clustering in Alarm Mining Category.

- Reliance on correlation rules: Many of the these systems are mainly relying on rule based correlation. Although few are using other methods of correlation and false alarm minimization, they are very elementary.
- Causal ordering based correlation, which finds relations between the alarms is a promising approach, however current tools seems to have not leveraged this completely barring one or two.
- Mining of alarms and hybrid methods for false alarm minimization are other two techniques which are promising and not yet commercialized.

14. Research questions

After studying various techniques for false alarm minimization we enlist following questions for the research community.

- *Evaluation on a common dataset:* We find most of the works evaluate their techniques on a local custom dataset. The performance of system analyzed in terms of false positives reduction ratio on a common dataset will help understand the usefulness.
- *Performance:* Many of the works found in literature chose to ignore performance aspects of their techniques. How much effort is required for the algorithm to execute is also one of the important characteristics.
- *Uniformity:* A uniform format to show the alarms and priority lists is required as most of the techniques use their own custom format and reporting structure. This will help using the processed information by other tools or logging into database for forensic reasons.
- *Realtime:* Many of the works do the correlation analysis on the offline data and that is one of the reason why they do not study performance aspects. We feel it is necessary to compare the techniques whether they can perform in real-time and help in controlling damages.
- *Incremental learning:* It is also important for any technique to adapt to changes that are happening in target networks, administrators interests, etc. A comparison in terms of how easy it is to adapt to these changes will help administrator chose a method based on her need.

15. Conclusion

In this paper, we have presented a survey of false alarm minimization techniques found in the literature. We have also provided a taxonomy of several techniques in Section 3. The figure presented in the section gives an overview of all techniques in a hierarchical format. In the section, we have also provided a comparison of our survey coverage with other known surveys. We have presented several techniques in this survey; first is signature enhancement where in few additional information is verified along with attack signature. Stateful signature-based approaches provided improved performance taking into account of the state of the network into account. Vulnerability signature-based approach overcomes all difficulties with application semantics and shown further improvement in terms of accuracy.

On the other hand several data mining techniques are also used for reducing false alarms such as alarm classification (which typically assumes a set of labeled alarms are available), alarm clustering (which divides the entire set of alarms into true or false alarm clusters), neural network (which also classify alarms after training) and frequent pattern mining (which mines the frequent item sets with a set of IDS alarms and their repeating pattern).

The fifth technique is alarm correlation which aggregates the number of alarms to predict the attack scenario. There are several approaches within these such as Multi-step correlation (which is based on the notion that there are several steps involved to stage an attack), knowledge based (which is based on extensive knowledge of the system being involved), complementary evidence based (which uses other type of known security component evidence), causal relation based (which integrates knowledge base correlation with statistical and temporal correlation techniques), fusion based (which implements prototype to understand multi-steps and also missing steps or alarms), attack graphs based (which identifies the path taken by the attacker) and rule based (which works by having if–then–else kind of rules).

Sixth technique is alarm verification which works based on the outcome of the attack and by verifying its impact on the system. Seventh technique is based on flow analysis which analyzes a set of alarms generated under normal and abnormal scenarios. Next technique is alarm prioritization which rates the alarms based on post assessment or some evaluation. Final technique is hybrid approach which combines two or more approaches together.

There are several commercial SIEM tools which handle security events generated in the networks. These events include IDS alarms too. A review of few prominent commercial SIEM tools is also provided in this paper. Some of the techniques described in this paper are used in these tools. We noticed that a majority of these tools still use rule based techniques for event correlation. Few other techniques like vulnerability signatures, alarm mining and hybrid approaches are interesting which can be incorporated to enhance the performance of these tools.

We have provided substantial analysis for every technique and approach and included all known recent papers with this survey (at the time of writing). Inspite of these all known techniques there are still issues to be addressed. We have also enlisted few of the research questions at the end. In our opinion future research need to address these research questions which will improve usability of the proposed techniques.

References

[1] J.P. Anderson, Computer Security Threat Monitoring, James P. Anderson Company, USA, 1980 (Technical report).
[2] D.E. Denning, An intrusion detection model, IEEE Trans. Softw. Eng. 13 (1) (1987) 222–232.
[3] K. Scarfone, P. Mell, Guide to Intrusion Detection and Prevention Systems, National Institute of Standards and Technology, 2007 (Technical report).
[4] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. 41 (2009) 15:1–15:58.
[5] T. Pietraszek, Using adaptive alert classification to reduce false positives in intrusion detection, in: RAID'04: Proceedings of the 7th International Conference on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, 2004, pp. 102–124.
[6] J.J. Treinen, R. Thurimella, Finding the needle: suppression of false alarms in large intrusion detection data sets, in: CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering, IEEE Computer Society, 2009, pp. 237-244.
[7] J.J. Treinen, R. Thurimella, Finding the needle: Suppression of false alarms in large intrusion detection data sets, in: CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering, IEEE Computer Society, 2009, pp. 237–244.
[8] S. Axelsson, The base-rate fallacy and its implications for the difficulty of intrusion detection, in: RAID '99: Proceedings of the 2nd International Symposium on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, 1999, pp. 1–7.
[9] S.T. Brugger, J. Chow, An Assessment of the Darpa IDS Evaluation Dataset Using Snort (2005), Technical report, <http://www.cs.ucdavis.edu/research/tech-reports/2007/CSE-2007-1.pdf>, University of California at Davis, 2007.
[10] G.C. Tjhai, M. Papadaki, S.M. Furnell, N.L. Clarke1, The problem of false alarms: Evaluation with snort and darpa 1999 dataset, in: TrustBus '99: Proceedings of the 13th USENIX System Administration Conference, Lecture Notes in Computer Science, 2008, pp. 139–150.
[11] R. Lipmann, J.W. Haines, D.J. Fried, J. Kobra, K. Das, The 1999 darpa off-line intrusion detection evaluation, Comp. Netw. 34 (4) (2000) 579–595.
[12] V. Paxson, Bro: a system for detecting network intruders in real-time, Comp. Netw. 31 (23–24) (1999) 2435–2463.
[13] U. Zurutuza, R. Uribeetxeberria, Intrusion detection alarm correlation: a survey, in: IADAT '04, Proceedings of the 2004 International Conference on Telecommunications and Computer Networks, IEEE, 2004, pp. 1–3.

[14] T. Limmer, F. Dressler, Survey of Event Correlation Techniques for Attack Detection in Early Warning Systems. Technical Report 01/08, University of Erlangen, Dept. of Computer, Science, April 2008.
[15] R. Sadoddin, A.A. Ghorbani, Alert correlation survey: framework and techniques, in: PST '06: Proceedings of the 2006 International Conference on Privacy, Security and Trust, ACM, 2006, pp. 1–10.
[16] F. Pouget, M. Dacier, Alert Correlation: Review of the State of the Art, Institute of Eurecom Corporate Communications Department, France, 2003 (Technical report).
[17] C.V. Zhou, C. Leckie, S. Karunasekera, A survey of coordinated attacks and collaborative intrusion detection, Comp. Sec. 29 (2010) 124–140.
[18] S. Salah, G. Maci Fernndez, J.E. Daz Verdejo, A model-based survey of alert correlation techniques, Comp. Netw. 57 (5) (2013) 1289–1317.
[19] S.O.A. Mamory, H.L. Zhang, A survey on ids alerts processing techniques, in: ISP'07': Proceedings of the 6th WSEAS International Conference on Information Security and Privacy, 2007, pp. 69–78.
[20] R. Sommer, V. Paxson, Enhancing byte-level network intrusion detection signatures with context, in: CCS '03: Proceedings of the 10th ACM Conference on Computer and Communications Security, ACM, 2003, pp. 262–271.
[21] F. Massicotte, M. Couture, Y. Labiche, Model-driven, network-context sensitive intrusion detection, in: MoDELS '07: Proceedings of the 10th International Conference on Model Driven Engineering Languages and Systems, Lecture Notes in Computer Science, 2007, pp. 61–75.
[22] S. Krishnamurthy, A. Sen, Stateful intrusion detection system (sids), in: ICIW '01: Proceedings of the 2nd IEE Conference on Information Warfare and Security, IEEE, 2001, pp. 1–10.
[23] G. Vigna, W. Robertson, V. Kher, R.A. Kemmerer, A stateful intrusion detection system for world-wide web servers, in: ACSAC '03: Proceedings of the 19th Annual Computer Security Applications Conference, IEEE Computer Society, 2003, p. 34.
[24] C. Kruegel, F. Valeur, G. Vigna, R.A. Kemmerer, Stateful intrusion detection for high-speed networks, in: S&P '02: Proceedings of the 2002 IEEE Symposium on Security and Privacy, IEEE Computer Society, 2002, pp. 285–294.
[25] M. Vallentin, R. Sommer, J. Lee, C. Leres, V. Paxson, B. Tierney, The nids cluster: Scalable, stateful network intrusion detection on commodity hardware, in: RAID '07: Proceedings of the 10th International Conference on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, 2007, pp. 1–20.
[26] H.J. Wang, C. Guo, D.R. Simon, A. Zugenmaier, Shield: vulnerability-driven network filters for preventing known vulnerability exploits, SIGCOMM Comp. Commun. Rev. 34 (2004) 193–204.
[27] D. Brumley, J. Newso, D. Song, H. Wang, S. Jha, Towards automatic generation of vulnerability-based signatures, in: S&P '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy, IEEE Computer Society, 2006, pp. 2–16.
[28] Z. Li, G. Xia, H. Gao, Y. Tang, Y. Chen, B. Liu, J. Jiang, Netshield: massive semantics-based vulnerability signature matching for high-speed networks, in: SIGCOMM '10: Proceedings of the 40th ACM SIGCOMM Conference, ACM, 2010, pp. 279–290.
[29] K. Julisch, M. Dacier, Mining intrusion detection alarms for actionable knowledge, in: SIGKDD '02: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2002, pp. 366–375.
[30] K. Julisch, Using Root Cause Analysis to Handle Intrusion Detection Alarms. Ph.D. Thesis, IBM Zurich Research Laboratory, Switzerland, 2003.
[31] K. Julisch, Clustering intrusion detection alarms to support root cause analysis, ACM Trans. Inform. Syst. Sec. 6 (4) (2003) 443–471.
[32] K. Julisch, Mining alarm clusters to improve alarm handling efficiency, in: ACSAC '01: Proceedings of the 17th Annual Computer Security Applications Conference, IEEE, 2001, pp. 12–21.
[33] R. Perdisci, G. Giacinto, F. Roli, Alarm clustering for intrusion detection systems in computer networks, Eng. Appl. Artif. Intell. 19 (2006) 429–438.
[34] E.H. Kim, M.S. Shin, K.H. Ryu, False alarm classification model for network-based intrusion detection system, in: IDEAL '04: Proceedings of the 2004 Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science, 2004, pp. 259–265.
[35] T. Pietraszek, A. Tanner, Data mining and machine learning – towards reducing false positives in intrusion detection, Inform. Sec. Tech. Rep. 10 (3) (2005) 169–183.
[36] S. Manganaris, M. Christensen, D. Zerkle, K. Hermiz, A data mining analysis of rtid alarms, Comp. Netw. 34 (4) (2000) 571–577.
[37] D. Parikh, T. Chen, Data fusion and cost minimization for intrusion detection, IEE Trans. Inform. Foren. Sec. 3 (3) (2008) 381–390.
[38] S. Benferhat, A. Boudjelida, K. Tabia, H. Drias, An intrusion detection and alert correlation approach based on revising probabilistic classifiers using expert knowledge, Int. J. Appl. Intell. 38 (4) (2013) 520–540.
[39] <http://www.logrhythm.com/>.
[40] Using Splunk Software as a SIEM – Tech Brief, 2013.
[41] C. Thomas, N. Balakrishnan, Performance enhancement of intrusion detection systems using advances in sensor fusion, in: Fusion'08: Proceedings of the 11th International Conference on Information Fusion, 2008, pp. 1671–1677.
[42] R. Sadoddin, A.A. Ghorbani, An incremental frequent structure mining framework for real-time alert correlation, Comp. Sec. 28 (2009) 153–173.
[43] M. Soleimani, A.A. Ghorbani, Critical episode mining in intrusion detection alerts, in: Proceedings of the Communication Networks and Services Research Conference, IEEE Computer Society, 2008, pp. 157–164.
[44] S. Templeton, K. Levitt, A requires/provides model for computer attacks, in: NSPW '00: Proceedings of the 2000 Workshop on New Security Paradigms, ACM, 2000, pp. 31–38.
[45] P. Ning, Y. Cui, S.D. Reeves, D. Xu, Towards automating intrusion alert analysis, in: 2003 Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, 2003, pp. 1–19.
[46] P. Ning, Y. Cui, D.S. Reeves, Analyzing intensive intrusion alerts via correlation, in: RAID'02: Proceedings of the 5th International Conference on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, 2002, pp. 74–94.
[47] P. Ning, Y. Cui, D.S. Reeves, D. Xu, Techniques and tools for analyzing intrusion alerts, ACM Trans. Inform. Syst. Sec. 7 (2) (2004) 274–318.
[48] S. Cheung, U. Lindqvist, M.W. Fong, Modeling multistep cyber attacks for scenario recognition, in: DISCEX '03: Proceedings of the 3rd International DARPA Information Survivability Conference and Exposition, IEEE, 2003, pp. 284–292.
[49] D. Yu, D. Frincke, A novel framework for alert correlation and understanding, in: ACNS '04: Proceedings of the 2004 International Conference on Applied Cryptoraphy and Network Security, Lecture Notes in Computer Science, 2004, pp. 452–466.
[50] F. Cuppens, F. Autrel, A. Miege, S. Benferhat, R.M. Ege, Correlation in an intrusion detection process, in: SECI '02: Proceedings of the 2002 International Conference on Security of Communications on the Internet, INRIA, 2002, pp. 153–183.
[51] F. Cupens, A. Miege, Alert correlation in a cooperative intrusion detection framework, in: S&P '02: Proceedings of the 2002 International Symposium on Security and Privacy, IEEE Computer Society, 2002, pp. 202–216.
[52] S.O. Al-Mamory, H. Zhang, Ids alerts correlation using grammar-based approach, J. Comp. Virol. 28 (3) (2009) 271–282.
[53] <http://www8.hp.com/in/en/software-solutions/>.
[54] <https://www.netiq.com>.
[55] TIBCO LOGLOGIC Security Event Manager, Datasheet, 2014.
[56] R.P. Goldman, W. Heimerdinger, S.A. Harp, C.W. Geib, V. Thomas, R.L. Carter, Information modeling for intrusion report aggregation, in: DISCEX 'II: Proceedings of the DARPA Information Survivability Conference and Exposition II, IEEE Computer Society, pp. 329–342.
[57] F. Valeur, G. Vigna, C. Kruegel, R.A. Kemmerer, A comprehensive approach to intrusion detection alert correlation, IEEE Trans. Depend. Secure Comput. 1 (3) (2004) 146–169.
[58] H.T. Elshoush, I.M. Osman, An improved framework for intrusion alert correlation, in: WCE'12:Proceedings of the 2012 World Congress on Engineering, 2012, pp. 1–6.
[59] S.C. Sundaramurthy, L.Z.X. Ou, Practical ids alert correlation in the face of dynamic threats, in: SAM'11: Proceedings of the 2011 International Conference on Security and Management, IEEE, 2011, pp. 1–7.
[60] G. Vigna, R.A. Kemmerer, Netstat: a network-based intrusion detection approach, in: ACSAC '98: Proceedings of the 14th Annual Computer Security Applications Conference, IEEE Computer Society, 1998, p. 25.
[61] G. Vigna, R.A. Kemmerer, Netstat: a network-based intrusion detection system, J. Comp. Sec. 7 (1) (1999) 37–71.
[62] J. Yu, Y.V.R. Reddy, S. Selliah, S. Kankanahalli, S. Reddy, V. Bharadwaj, Trinetr: An intrusion detection alert management system, in: WETICE04: Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE, 2004, pp. 235–240.
[63] RSA enVision Platform, Compliance and Security Information Management, RSA Solution Brief, 2013.
[64] <http://www.mcafee.com/in/products/advanced-correlation-engine.aspx>.
[65] IBM Security QRadar SIEM, Data-Sheet, 2014.
[66] H. Debar, A. Wespi, Aggregation and correlation of intrusion-detection alerts, in: RAID '00: Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection, Springer-Verlag, 2001, pp. 85–103.
[67] T. Chyssler, S.N. Tehrani, S. Burschka, K. Burbeck, Alarm reduction and correlation in defence of IP networks, in: WETICE '04: Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE Computer Society, 2004, pp. 229–234.
[68] X. Qin, A Probabilistic-Based Framework for INFOSEC Alert Correlation. Ph.D. dissertation, College of Computing Georgia Institute of Technology, USA, 2005.
[69] F. Maggi, S. Zanero, On the use of different statistical tests for alert correlation: short paper, in: Proceedings of the 10th International Conference on Recent Advances in Intrusion Detection, RAID'07, Springer-Verlag, pp. 167–177.
[70] J. Viinikka, H. Debar, L. Mé, A. Lehikoinen, M. Tarvainen, Processing intrusion detection alert aggregates with time series modeling, Inform. Fus. 10 (2009) 312–324.
[71] H. Ren, N. Stakhanova, A.A. Ghorbani, An online adaptive approach to alert correlation, in: DIMVA'10: Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Lecture Notes in Artificial Intelligence, 2010, pp. 153–172.
[72] N. Carey, A. Clark, G.M. Mohay, Ids interoperability and correlation using idmef and commodity systems, in: ICICS '02: Proceedings of the 4th International Conference on Information and Communications Security, Lecture Notes in Computer Science, 2002, pp. 252–264.
[73] C. Feng, J. Peng, H. Qiao, J.W. Rozenblit, Alert fusion for a computer host based intrusion detection system, in: Proceedings of the 14th Annual IEEE

International Conference and Workshops on the Engineering of Computer-Based Systems, IEEE Computer Society, 2007, pp. 433–440.
[74] S. Noel, S. Jajodia, Optimal ids sensor placement and alert prioritization using attack graphs, J. Netw. Syst. Manage. 16 (2008) 259–275.
[75] S. Noel, S. Jajodia, Advanced vulnerability analysis and intrusion detection through predictive attack graphs, in: AFCEA'09: Critical Issues in C4I in Armed Forces Communications and Electronics Association Solutions Series, IEEE, 2009, pp. 1–10.
[76] AlienVault Unified Security Management, Data-Sheet, 2013.
[77] D. Bolzoni, C. Bruno, E. Sandro, ATLANTIDES: an architecture for alert verification in network intrusion detection systems, in: LISA'07: Proceedings of the 21st Conference on Large Installation System Administration Conference, USENIX Association, 2007, pp. 1–12.
[78] J. Zhou, A.J. Carlson, M. Bishop, Verify results of network intrusion alerts using lightweight protocol analysis, in: Proceedings of the 21st Annual Computer Security Applications Conference, IEEE Computer Society, 2005, pp. 117–126.
[79] C. Kruegel, W. Robertson, Alert verification – determining the success of intrusion attempts, in: DIMVA '04: Proceedings of 1st Workshop the Detection of Intrusions and Malware and Vulnerability Assessment, Lecture Notes in Computer Science, 2004, pp. 1–14.
[80] C. Kruegel, W. Robertson, G. Vigna, Ausing alert verification to identify successful intrusion attempts.
[81] A.D. Todd, R.A. Raines, R.O. Baldwin, B.E. Mullins, S.K. Rogers, Alert verification evasion through server response forging, in: RAID'07: Proceedings of the 10th International Conference on Recent Advances in Intrusion Detection, 2007, pp. 256–275.
[82] J. Viinikka, H. Debar, Monitoring ids background noise using ewma control charts and alert information, in: RAID '04: Proceedings of the 2004 International Symposium on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, 2004, pp. 166–187.
[83] P.A. Porras, M.W. Fong, A. Valdes, A mission-impact-based approach to infosec alarm correlation, in: RAID 02', Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, 2002, pp. 95–114.
[84] C. Mu, H. Huang, S. Tian, Intrusion detection alert verification based on multi-level fuzzy comprehensive evaluation, in: CIS '05: Proceedings of the 2nd International Conference on Computational Intelligence and Security, Lecture Notes in Artificial Intelligence, 2005, pp. 9–16.
[85] K. Alsubhi, E.A. Shaer, R. Boutaba, Alert prioritization in intrusion detection systems, in: NOMS '08: Proceedings of the 11th IEEE/IFIP Network Operations and Management Symposium, IEEE, 2008, pp. 33–40.
[86] G.P. Spathoulas, S.K. Katsikas, Reducing false positives in intrusion detection systems, Comp. Sec. 29 (2) (2010) 35–44.
[87] SEBASTIAN KLFT, Alarm Management for Intrusion Detection Systems Prioritizing and Presenting Alarms From Intrusion Detection Systems, Master of Science Thesis, Chalmers University of Technology, University of Gothenburg, 2010.
[88] N. Hubballi, S. Biswas, S. Nandi, Network specific false alarm reduction in intrusion detection, Sec. Commun. Netw. 4 (2011) 1339–1349.
[89] F. Gagnon, F. Massicotte, B. Esfandiari, Using contextual information for ids alarm classification (extended abstract), in: DIMVA '09:Proceedings of the 6th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Lecture Notes in Computer Science, 2009, pp. 147–156.
[90] C. Abad, Y. Li, K. Lakkaraju, X. Yin, W. Yurcik, Correlation between netflow system and network views for intrusion detection, in: WLA '04: Workshop on Link Analysis, Counter-Terrorism, and Privacy, ACM, 2004.
[91] Junifer Networks, Accurate attack detection, in: Junifer Networks Datasheet, 2005, pp. 1–6.
[92] S.T. Eckmann, G. Vigna, R.A. Kemmerer, Statl: an attack language for state-based intrusion detection, J. Comp. Sec. 10 (2000) 1–16.
[93] S.O. Al-Mamory, H. Zhang, New data mining technique to enhance ids alarms quality, J. Comp. Virol. 28 (2) (2010) 43–55.
[94] S.O. Al-Mamory, H. Zhang, Intrusion detection alarms reduction using root cause analysis and clustering, Comp. Commun. 32 (2) (2009) 419–430.
[95] K.H. Law, L.F. Kwok, Ids false alarm filtering using knn classifier, in: WISA'04: Proceedings of the Workshop on Information Security Applications, Lecture Notes in Computer Science, 2004, pp. 114–121.
[96] C. Dey, Reducing Ids False Positives Using Incremental Stream Clustering (isc) Algorithm, M.Sc. Thesis, Royal Institute of Technology, Sweden, 2009.
[97] G.C. Tjhai, S.M. Furnell, M. Papadaki, N.L. Clarke, A preliminary two-stage alarm correlation and filtering system using som neural network and k-means algorithm, Comp. Sec. 29 (3) (2010) 712–723.
[98] X. Zhuang, D. Xiao, X. Liu, Y. Zhang, Applying data fusion in collaborative alerts correlation, in: ISCSCT '08:Proceedings of the 2008 International Symposium on Computer Science and Computational Technology, vol. 02, IEEE Computer Society, 2008, pp. 124–127.
[99] J. Chang, J. Yu, Y. Pie, MSIFS: a multiple source-based security information fusion system, in: IICCIIS'10: Proceedings of the 2010 International Communications and Intelligence Information Security Conference, IEEE, 2010, pp. 215–219.
[100] M. Wood, M. Erlinger, Intrusion Detection Message Exchange Requirements, draft-ietf-idwg-requirements-05.txt, 2007.
[101] H. Debar, D.A. Curry, S.F. Benjamin, The intrusion detection message exchange format (idmef), Request for Comments 4765 (2007).
[102] A.B. Mohamed, N.B. Idris, B. Shanmugum, Article: an operational framework for alert correlation using a novel clustering approach, Int. J. Comp. Appl. 54 (12) (2012) 23–28.
[103] J. Zhou, M. Heckman, B. Reynolds, A. Carlson, M. Bishop, Modeling network intrusion detection alerts for correlation, ACM Trans. Inform. Syst. Sec. 10 (1) (2007) 4.
[104] F. Cupens, R. Ortalo, Lamda: a language to model a database for detection of attacks, in: RAID '00: Proceedings of the 2000 International Symposium on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, 2000, pp. 197–216.
[105] A. Valdes, K. Skinner, An approach to sensor correlation, in: RAID '00: Proceedings of the 3rd International Symposium on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, 2000, pp. 1–11.
[106] Sourcefire, Sourcefire ips the foundation of the sourcefire 3d system, White Paper, 2010.
[107] B. Morin, H. Debar, M. Ducassé, M2d2: a formal data model for ids alert correlation, in: RAID'02: Proceedings of the 5th International Conference on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, 2002, pp. 115–137.
[108] A. Sharmaand, Z. Kalbarczyk, R.K. Iyer, J. Barlow, Analysis of credential stealing attacks in an open networked environment, in: NSS'11: Proceedings of the 2011 International Conference on Networks and System Security, IEEE Computer Society, 2010, pp. 144–151.
[109] A. Sharmaand, Z. Kalbarczyk, J. Barlow, R.K. Iyer, Analysis of security data from a large computing organization, in: DSN'11: Proceedings of the 2011 International Conference on Dependable Systems and Networks, IEEE, 2011, pp. 506–517.
[110] CISCO, Firewall Intrusion Detection System Signature Enhancements, CISCO IOS Release, 2009.
[111] X. Qin, W. Lee, Discovering novel attack strategies from infosec alerts, in: ESORICS '04: Proceedings of the 9th European Symposium on Research in Computer Security, Lecture Notes in Computer Science, 2004, pp. 439–456.
[112] H.K. Pao, C.H. Mao, H.M. Lee, C.D. Chen, C. Faloutsos, An intrinsic graphical signature based on alert correlation analysis for intrusion detection, J. Inform. Sci. Eng. (2012) 243–262.
[113] P. Ning, D. Xu, C.G. Healey, R.S. Amant, Building attack scenarios through integration of complementary alert correlation methods, in: NDSS'04: IN Proceedings of the 11th Annual Network and Distributed System Security Symposium, ACM, 2004, pp. 97–111.
[114] S. Roschke, F. Cheng, C. Meinel, A new alert correlation algorithm based on attack graph, in: CISIS'11: Proceedings of the 4th International Conference on Computational Intelligence in Security for Information Systems, Springer-Verlag, 2011, pp. 58–67.
[115] M. Albanese, S. Jajodia, S. Noel, Time-efficient and cost-effective network hardening using attack graphs, in: DSN'1: Proceedings of the 43rd Conference on Dependable Systems and Networks, IEEE, 2012, pp. 1–12.
[116] C. Mu, X.J. Li, H. Huang, S. Tian, Online risk assessment of intrusion scenarios using d-s evidence theory, in: ESORICS '08: Proceedings of the 13th European Symposium on Research in Computer Security, Lecture Notes in Artificial Intelligence, 2008, pp. 35–48.
[117] A.D.J. Valdes, M.W. Fong, P.A. Porras, Prioritizing Bays Network Alerts-Sri International, US Patent-7379993 B2, May-27, 2008.
[118] M. Roesch, Snort – lightweight intrusion detection for networks, in: LISA '99: Proceedings of the 13th USENIX System Administration Conference, USENIX Association, 1999, pp. 229–238.
[119] <http://www.sectools.org/vuln-scanners.html>.
[120] <http://www.nessus.org>.
[121] <http://www.gfi.com>.
[122] <http://www.nmap.org>.
[123] Retina <http://www.eeye.com/products/retina.aspx>.
[124] Juniper Networks, Idp Series Intrusion Detection and Prevention Appliances, Product Category Broucher, 2009.
[125] K. Lakkaraju, W. Yurcik, A.J. Lee, NVisionIP: Netflow visualizations of system state for security situational awareness, in: vizSEC'04: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, ACM, 2004, pp. 65–72.
[126] X. Yin, W. Yurcik, M. Treasterl, Y. Li, K. Lakkaraju, Visflowconnect: netflow visualizations of link relationships for security situational awareness, in: VizSEC/DMSEC '04: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, ACM, New York, 2004, pp. 26–34.
[127] <http://www.ndm.net/siem/main/rsa-envision-siem>.
[128] <http://www.ndm.net/siem/arcsight/arcsight-esm>.
[129] <http://www.ndm.net/siem/logrhythm-siem-20>.
[130] QRadar Security Intelligence Platform Appliances. QRadar-Appliance-Family-Data-Sheet, 2013.
[131] NUTANIX, Splunk Reference Architecture, Splunk on Nutanix, 2013.
[132] Tenable Log Correlation Engine, Datasheet, 2013.
[133] LogCenter RecoemndedInstallation Sizing, Datasheet, 2013.
[134] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (January 2008) 107–113.
[135] <http://hadoop.apache.org/>.