Methods for post-processing of alerts in intrusion detection: A survey

Georgios P. Spathoulas[1,2], Sokratis K. Katsikas[3]

[1] Department of Computer Science and Biomedical Informatics, University of Central Greece,
2-4 Papasiopoulou St., Lamia, GR-35100, Greece, e-mail: gspathoulas@ucg.gr
[2] Department of Digital Systems, School of Information and Communication, University of Piraeus,
150 Androutsou St., Piraeus, GR-18532, Greece, e-mail: gspathoulas@unipi.gr
[3] Department of Digital Systems, School of Information and Communication, University of Piraeus,
150 Androutsou St., Piraeus, GR-18532, Greece, e-mail: ska@unipi.gr

Abstract—Intrusion detection is an important protection tool for computer systems and networks. In recent years it has become an essential piece in the IT security infrastructure of large organizations. Even though intrusion detection systems are installed in an increasing rate, they are often misused as the quality of alerts they produce is not satisfactory. High alert volume, high false positives rate and low level of information are the main reasons that security analysts cannot take full advantage of intrusion detection alert-sets. The aim of this survey is to summarize intrusion detection alerts' post-processing research, which is categorized in false positives reduction, alerts' correlation and visualisation. The most important efforts in the field are analyzed, while all recent methods are presented. Finally the present and the future of alerts post-processing research field is discussed.

Keywords—Intrusion detection, alerts, post-processing, false positives reduction, correlation, visualization.

1. Introduction

One of the tools in the hands of security analysts' is intrusion detection systems. These systems detect possible intrusions and produce alerts in order to notify the analyst of the intrusion. There are many types of such systems, as detection technique and scope of the protected system may vary. Host based systems protect critical hosts, while network systems can protect a whole network. Signature-based systems use predefined intrusion profiles and try to match the activity of the protected system to them, while anomaly-based systems search for important deviations from normality in the system's activity, and characterize these as intrusions.

An important problem in the intrusion detection field, regardless of the system type, is the low quality of the produced alert-sets, due to which intrusion detection systems may become unusable. The volume of alerts is usually difficult to manage, while false positives and false negatives are always present. As a result generally, alert-sets as produced by intrusion detection systems are hard for security analysts to utilize. In order to overcome this, a lot of research has been focused on improving the quality of alerts and specifically on the post-processing of alerts.

This work paper presents recent research work on post-processing intrusion detection alerts. It is structured as follows: Section 2 presents the different aspects of the alert quality problem and the need for post-processing, while section 3 presents important and pioneering research conducted in the early years of the last decade. The next three sections review recent work: section 4 reviews work concerning reducing false positives, section 5 reviews work that concerns correlating relevant alerts and section 6 reviews work that concerns alert visualization. Finally, in section 7 we discuss limitations of existing methods and future research possibilities in post-processing alerts.

2. The need for post-processing of intrusion detection alerts

As soon as intrusion detection research matured and the first real world implementations of such systems were installed, the problem of the low quality of the resulting alert-set became evident. In theory intrusions were detected at a high rate, but the alert-sets produced by the intrusion detection systems were inappropriate for use in an environment where instant reaction is critical. Security analysts had to dig through huge alert-sets of low quality to find indications of intrusions. The difficulty in deciding whether an intrusion was really occurring made them to either react with delay or even to not react at all to real intrusions that were indeed detected by the intrusion detection systems. The main aspects of this problem of low quality of alerts are:

- **High volume:** Normally, intrusion detection systems protect complicated systems and networks, whose utilization is relatively high; thus, high volumes of data (network traffic or system calls) are examined for possible malicious activity. This produces large volumes of alerts. In most of the cases it is impossible for the analyst to read a real world alert-set in an alert by alert fashion, as alerts are produced at a rate much higher than the rate in which she can read them.
- **High false positives rate:** Apart from being huge in volume a real world alert-set consists mainly of false alerts, i.e. alerts that do not correspond to real intrusion incidents. This mainly happens because intrusion detection systems try to achieve high detection rates (percentage of true intrusions detected), so their sensitivity is set at relatively high levels.
- **Low level of alerts :** Alerts correspond to low level events in a system (e.g. to an IP packet or to a system call). Attempted intrusions are higher level events and they usually produce multiple different alerts. This difference in the level between events and alerts makes it hard to infer useful information when reading an alert-set.

In order to minimize these defects, several researchers have employed methods of post-processing intrusion detection alerts. These methods fall into three main methodologies; reduction of false positives clustering; and alternative representation techniques.

3. Early important work on post-processing of intrusion detection alerts

In the last two decades some serious research work on post-processing alerts has been conducted quality. In this Section the most important research efforts in the field are presented.

3.1. Defining alerts similarity

In 2001 Valdes and Skinner proposed using probabilistic similarity between alerts as a means to post-process them [1]. To this end, they defined a method for calculating similarities. They calculate similarity between two alerts as a meter of overlapping between their features. Since then many methods that use this similarity approach or try to enhance it have been proposed.

In this approach alerts for which there is a relevant match are aggregated. For each different alert attribute an appropriate similarity function is defined. Additionally, an expected similarity value is calculated, which in practice is a weight that is later used to calculate the overall similarity. A minimum match specification is also incorporated, that unconditionally rejects a match if any feature similarity is lower than the minimum specified value. For each new alert, the similarity to all existing meta alerts is computed taking into account attribute similarities along with the corresponding expected similarities. The alert is then merged with the best matching meta alert, as long as their similarity is above a threshold value.

An experimental process has been conducted, with a simulated real world network. Normal traffic was artificially generated and at the same time the designed attack was executed. The intrusion detection sensors used were EMERALD eBayes and eXpert-Net. The correlation procedure has achieved a reduction of false alerts at one-half to two-thirds with regards to the initial alert-set.

The concept of combining results of similarity functions for each attribute of alerts, to calculate an overall similarity has influenced the work of other researchers [2],[3],[4].

3.2. Discriminating between aggregation and correlation

At about the same time, Debar and Wespi [5] presented the first analytical descriptions of alert aggregation and correlation procedures. They discuss an overall intrusion detection post-processing architecture and their well defined approach remains valid until today, as most methods after theirs have discriminated between aggregation and correlation.

They highlight the most important problems in intrusion detection alert-sets as:

- Intrusion detection systems provide the operator with a large number of alerts; the operator then has difficulties coping with the load.
- Attacks are likely to generate multiple related alerts and it is not easy for operators to logically group them.
- Intrusion detection systems are likely to generate many false alerts, false positives or false negatives.
- Intrusion detection system architectures, at the time, made it difficult to achieve large-scale deployment.

In order to solve these problems, the authors proposed an architecture that consists of multiple detection probes, the outputs of which are fed to aggregation and correlation components.

In the aggregation component the algorithm groups events together according to certain criteria. The aim is to discard multiple identical alerts at the sensor level.

In the correlation component the algorithm creates correlation relationships between related events according to explicit rules. Once events are in a relationship, they are considered as part of the same attack and are processed together. The authors define two kinds of correlation relationships between events: duplicates and consequences.

The detection of duplicates relies on the provision of common information by different intrusion detection sensors. Duplicates are alerts referring to exactly the same event. Consequence chains are sets of alerts linked in a given order, where the link must occur within a given time interval. Consequences are alerts that correspond to consecutive related alerts.

A usage example is given, but thorough experiments are not carried out.

3.3. Reconstructing attack scenarios

In [6] the motivation is to provide a framework for constructing attack scenarios through alert correlation, using prerequisites and consequences of intrusions. The approach is based on the observation that alerts correspond to different stages of an attack scenario, with the earlier stages preparing for the later ones. The same idea has been reused in recent years in many research efforts [7],[8],[9].

The authors proposed a formal framework to represent alerts along with their prerequisites and consequences, and developed a method to correlate related alerts. In this framework they define hyper alerts types, which are composed by the intrusion type, the prerequisites and the post conditions of the intrusion. The prerequisites of an intrusion are the necessary conditions for the intrusion to be successful, while the consequences of an intrusion are its possible outcomes. They also developed an off-line tool on the basis of the formal framework, which tries to correlate alerts, by combining post conditions with prerequisites. Specifically, the tool examines each alert and tries to discover possible combinations of its post conditions with the prerequisites of alerts with timestamps in a specific time window which comes later than the time stamp of the alert being examined.

The authors conducted experiments that demonstrated the potential of their method in correlating alerts. While the method is based on manually defining prerequisites and post conditions for all possible attack types, a fact that reduces flexibility and ability to deal with new attack types, the idea of connecting alerts in a logical and chronological manner was important for later post-processing of alerts research.

3.4. A complete approach

Perhaps the most influential work in post-processing of alerts was presented in [10]. The authors in [10] have implemented a complete system that tackles most aspects of post-processing of alerts and have conducted experiments on different datasets to prove the validity of their assumptions.

They provided a detailed analysis of the problem and designed a set of components that focus on different aspects of the overall correlation task. First, a normalization component transforms all alerts to a standardized format, understood by all correlation components. Next, a preprocessing component deals with attributes of alerts that sensors may have omitted and supplies relevant values, as accurate as possible. These attributes may be required for the functioning of other components in sequel. The fusion component is responsible for combining alerts that represent independent detections of the same attack instance by different intrusion detection systems. In a system with multiple sensors, identical alerts may be an important problem to solve. A verification component determines the possible success of the attack which each alert corresponds to; this information is used by the correlation components down the line. Verification is performed either by using passive techniques, such as gathering data for the network in advance, or by using active techniques, such as looking for attack success evidence after the alert has been recorded.

The thread reconstruction component identifies combinations of attacker and target through all alerts, in order to discover series of alerts that refer to attacks launched by a single attacker against a single target. This component is important as it can associate network-based alerts with host-based alerts, both related to the same attack. The task of the focus recognition component is to identify hosts that are either the source or the target of a significant number of alerts. These hosts are likely to be related to a denial-of-service attack or to a port scanning attempt. The multistep correlation component identifies predefined common attack patterns such as island-hopping attacks. Finally, the impact analysis component determines the impact of the detected attacks on the operation of the network being monitored; this information is eventually used by the prioritization component, which assigns an appropriate priority to every alert.

Besides designing a robust system, the authors extensively researched data-sets available at the time, and utilized all of them to experimentally test their system.

4. Reducing false positives

As discussed in section 2, intrusion detection alert-sets are characterized by high false positives rate. Various numerous methods have been proposed to cope with this problem. In this section the latest research efforts to reduce false positives in intrusion detection are presented.

4.1. Considering initial classification inadequate

The author in [11] suggests that the actual intrusion detection systems are inadequate, as he proves that checking the TTL of packets that produced the alerts helps reducing significantly the false positives rate. The proposed method is based on clustering the produced alerts on the basis of their TTL values. Experiments on various data-sets show that false positives are included in specific clusters and that it is then easy to discard them. This work has been extended in [12], whereby apart from false positives, redundant alerts are also filtered out through clustering procedures.

4.2. Looking at neighboring alerts

The authors in [13] take advantage of mixture models in order to discriminate between true and false positives. They compare the characteristics of each alert to the characteristics of previous ones. It is expected that a true alert will differentiate from its precedents. In this way the alert is classified to the intrusion or to the non-intrusion set. Additionally the protected system is checked for vulnerabilities relevant to each alert, in order to characterize it as critical or not.

While the method seems interesting, it is not well defined and the reader is not convinced of its validity.

The authors in [14] also examine neighboring alerts, to decide on the validity of each alert. They calculate the relevant correlation, and try to identify false alerts along with duplicate ones (alerts that tend to reappear multiple times through the dataset). Apart from that, they also use an ensemble-based adaptive learner which, given the expertise feedback, is capable of adapting to environmental changes through automatic tuning. The learner remains effective even if the protected network changes. The implementation is tested by using both the DARPA and a private data set. The method requires the intervention of the security analysts; this makes it inapplicable in real world, large scale networks.

The alert-set itself is also utilized in [15]. The method proposed therein is based on the calculation of reputation for alerts. The reputation relates to the probability that these alerts are true. It is calculated from the false positives rate of alerts concerning the same IP address or sharing the same signature. The performance of the method is validated through experiments that show significant reduction in the false positives rate. The limitation of the method is that in order to calculate the reputation, the validity of previous alerts needs to be known.

In [16] a post-processing filter is presented. The authors propose a filter consisting of four different components, each of which examines various parameters of each alert in relation to its neighbors and calculates the probability that this alert is true (i.e. that it corresponds to a true intrusion event). The examined parameters are the percentage of neighbors sharing the same signature and having similarities in source and destination IPs; the deviation in the frequency with with which a signature appears and the tendency of a signature to produce false positives. The probabilities calculated by all components are then combined, to produce a final verdict for each alert. Finally, comparison against a threshold value is used to discard false alerts. Experiments conducted show that the filter performs well at least with the DARPA data-set.

4.3. Training required

The authors in [17] propose the use of an adaptive false alert filter that incorporates some of the most common machine learning techniques, such as K-nearest neighbors; Decision trees; and Support vector machines. The results of each method are continuously monitored. The filter examines each algorithm's performance every hour and chooses the best of them to be used until the next evaluation. The filter seems to perform well, but there is an important drawback, as the comparison of the algorithms' performance is based upon the labeled Snort data-set (the validity of alerts can be determined from the DARPA documentation). In a real world scenario no such labels exist.

In [18] the main assumption exploited is that false positives are triggered by causes that are frequent in a specific network. A training phase is described, where the frequency of values for attributes of alerts are calculated. These frequencies are stored in hash tables. They are normalized by weight values and used to determine if a future alert is similar to alerts that frequently appear or if it is significantly different from them. The threshold value that is used to discard frequent alerts is also decided upon during training phase.

4.4. Considering attacks as anomalies

In [19] valid alerts are considered as anomalies in an alert-set mainly consisting of false positives. The authors built profiles of usual false positives, for a given protected network. The idea is to use anomaly detection techniques on the produced alert-set, to discriminate true alerts as alerts that are characterized by an important deviation from these profiles. Three different algorithms are used and compared in terms of achieved false positives reduction, given the fact that they do not filter out any true positive. The rationale of this paper seems interesting and promising, but the choice of the threshold that discards alerts would be difficult to make in a real world scenario, where no evidence regarding the validity of the alerts exists.

The same idea is more or less found in [20]. The authors therein state that there are root causes for each group of similar alerts. If the protected system's administrator can discriminate root causes relevant to intrusions from the ones relevant to benign activity, then she could easily reduce the false positives rate by discarding the alerts produced by non intrusive root causes. They propose to use clustering in order to create clusters of similar alerts and then to characterize each of these clusters according to an assumed root cause. Future alerts can be characterized by the cluster they are closer to. An obvious drawback of the proposed system is that it is semi-automatic, as it will always require human intervention in order for root causes to be discovered and characterized as nominal activity or intrusion.

In [21] the idea is to reduce the overall number of alerts and by doing so to reduce the false positives rate. K-means clustering is used to identify main clusters in a huge population of alerts for a specific network, while outliers are ignored. The authors state that if a future alert could be categorized into one of these clusters it would be a strong indication that the specific alert concerns nominal traffic. Upon this hypothesis, they propose to completely ignore such alerts. While they provide proof for the high percentage of alerts that could be ignored in this way (resulting in a much smaller alert-set), they do not provide enough evidence on the validity of their assumption, that the ignored alerts are indeed relevant to nominal traffic.

4.5. Getting feedback

A general framework is proposed in [22] that enables the feedback of false positives occurring in results to be fed back into the monitoring process. This way the policy implemented by the intrusion detection system can be altered accordingly to the false positive rate. The framework proposed therein is a solid base on which methods for reconfiguring monitoring policies can be based, but there are important issues that should be addressed in order for the framework to be used in a real world scenario. Specific adjustments should be made to the framework itself as it is strongly coupled to the intrusion detection system used, the procedure chosen for checking the validity of alerts and the method for evaluating the performance of the monitoring policy in use.

4.6. Working under uncertainty

The authors in [23] deal with the aggregation of alerts, in order to reduce false positives, but in terms of anomaly detection systems. Alerts, produced by signature based systems, contain useful information for alert correlation methods, such as the relevant signature or the class of the attack. On the other hand, anomaly detection systems produce less information for the attack and the correlation process is more difficult. In this paper the use of fuzzy sets is proposed, in order to avoid missing alerts due to fuzziness issues. The main information to base aggregation in anomaly-based systems is timestamps. Problems may appear in this aggregation procedure due to system latency or wrong sharp threshold values. The authors discuss the criteria required to compute the time distance between alerts and to define threshold values by taking into account the uncertainty factor. The general intrusion detection feasible data-set problem is thoroughly discussed and a framework for evaluating fusion methods is presented.

The authors in [24] propose the use of a fuzzy inference system, which filters out false positives, without missing on any of the detected attacks. For each alert a meta-alert is calculated, the fields of which are relevant to specific statistical observations about the alert and its relation to the other alerts that are close in time to it. The meta-alert is led to a fuzzy inference system which uses predefined membership functions to transform each one of its attributes to membership degrees. Logical if-then rules are incorporated to calculate a degree of membership at the output of the system, which corresponds to whether the alert is true. By the use of a threshold value this degree can be used to classify alerts as true or false. The system has been tested against the DARPA dataset and has exhibited a significant reduction (83%) of false positives, while it filters in alerts for all the attacks that really occurred.

4.7. *Considering network's vulnerabilities*

The authors in [25] propose to filter out false positives by taking into account the vulnerabilities of the protected systems. They assume that every alert that concerns an ineffective, to the protected network, attack can be treated as a false positive. Even if there is an ongoing intrusion attempt, it will be unsuccessful as the required vulnerability is missing. The proposed method uses vulnerability scanners to create profiles of existing vulnerabilities. Alerts are then correlated to these profiles and the resulting distance vectors are used to classify each alert as true or false. A back-propagation neural network has been trained on vectors that belong to alerts known to be ineffective. Then, this neural network classifies new alerts as effective or ineffective, filtering out the latter ones. The experimental results on a custom data-set indicate that the accuracy of the intrusion detection system has been vastly improved.

5. Correlating alerts

Perhaps the most research-intensive field in intrusion detection post-processing is the correlation of alerts. Due to the multiple feasible approaches to the problem, a large volume of relevant papers exist in the literature. In this section we present the most recent works in the field. Some authors extend work that was done previously, while others propose more innovative solutions.

5.1. *Methods in their early stages*

In [26] a simple alert clustering scheme is proposed, in order to reduce the number of alerts. The alerts' attributes that are examined are the alert's signature; the destination IP; and the time stamp. While the clustering algorithm used is not described in detail, each produced cluster is tagged with an attack type at the end of the procedure. This research work seems to be in progress hence, criticizing its results is immature.

An iterative clustering procedure is presented in [27]. The ISODATA algorithm creates clusters of alerts in an iterative manner. The functioning and the finalizing circumstances of the algorithm are based on arbitrarily valued parameters. The experimental results indicate a reduction of the number of alerts, albeit without justifying the validity of the produced aggregations. The method does not seem to contribute much to recent intrusion detection research.

The methodology described in [28], aims to aggregate intrusion detection alerts in a performance-efficient manner, in order to be applicable in an online scenario. The authors regard attack instances as random processes producing alerts and they try to model these processes, using approximate maximum likelihood parameter estimation techniques. While they provide a general description of their theory, there is no analysis of the proposed method.

5.2. *Using correlated alerts to reduce false positives*

The authors in [29] have incorporated the results of correlating alerts, to reduce false positives. They use the Self Organising Maps algorithm to create clusters of alerts, triggered by the same security events, in an unsupervised manner. After that, they use the K-Means algorithm to further classify the clusters as true or false. The output of the SOM algorithm is fed as input to the K-means algorithm. The experiments carried on both the DARPA dataset and on a private University of Plymouth dataset prove that two stage clustering is efficient. Another advantage of the proposed method is that the graphical representation of clusters produced by the SOM algorithm may be representative of the relations between correlated alerts.

In [30] frequent itemset mining is used in order to discover alerts that are frequent and characterize them as normal. Alerts that look like unusual (interesting) are isolated and promoted as possible intrusions. The authors have collected a private alert-set; a thorough examination of the properties of these alerts has indicated that strong recurring patterns exist in the alert-set. The frequent itemset mining algorithm has been adapted to address problems of the specific domain. For example, unusual and intensive short-term malicious network activity may produce too many alerts and thus trigger relative patterns that will classify similar future activity as normal. On the other hand, too generic produced patterns may be inappropriate for alert classification due to over-generalization issues. The experimental results showed that the classification of alerts as interesting has achieved high rates of precision, as most of the intrusion-relevant alerts were in that category, while the size of the alert-set was substantially reduced.

5.3. Socialization between intrusion detection nodes

An interesting approach for optimising collaboration among intrusion detection nodes is presented in [31]. It is based on the assumption that each node can communicate with other nodes and it can appraise their trustworthiness. In this way, each node can get information about ongoing attacks from other nodes and evaluate it according to the confidence it has for them. The authors describe a formal mathematical model that dictates how nodes get to know other nodes, how they manage trustworthiness through time, how they send consultation requests and, finally, how they decide on the validity of alerts based on aggregating advice from other nodes through a Bayesian approach. This method does not deal with correlation in the usual manner, but it provides a new aspect on how alerts produced by different intrusion detection systems can be combined.

The authors in [32] propose a framework that is based on an hierarchical view of the protected system. They first define ontologies for all the involved parts and then describe a model through which correlation of alerts is achieved even though the alerts may concern various kinds of assets of the protected system or they have been generated by various types of sensors. Moreover, a trustworthiness factor for each of these combinations is taken into account in order to produce results that approximate the true security status of the protected system, as accurately as possible. A limited manual experimental procedure is described in order to assess the efficiency of the proposed method, while more extensive tests are required, as the likely diversity of sensors or systems may prove the framework inapplicable.

An important requirement for successful alert correlation between different intrusion detection systems is the existence of a common representation for alerts. In [33] a well defined representation model is presented, based on the first-order logic formalism. Apart from describing a representation model for intrusion detection alerts, the authors have also tried to formalise representations for all other important entities in intrusion detection context. They propose representation schemes for hosts, software products, vulnerabilities, attack classes, intrusion detection systems, events, messages etc. The model is interesting and its use seems promising. Further testing has to be done in order to evaluate if this theoretical model is applicable to real word alert correlation scenarios.

5.4. Decentralizing alert correlation

A well defined methodology is proposed in [34], in order to cope with the correlation of alerts produced by intrusion detection systems scattered all over the world. The authors define an alert attribute pattern scheme, that is used to efficiently represent alerts. They use this scheme to commit a two stage correlation, one locally for each intrusion detection system and one for the global system. The two stages of correlation ensure that there will not be any computation overhead issues. A methodology for automatically setting the local and global thresholds is defined. Two different models of processing are examined, one with a central server and the other completely decentralized, built on a peer-to-peer architecture. The evaluation of the algorithm is also concrete, as it takes into account the geographical location of the sensors and it calculates the corresponding communication overheads. In general, the authors have made an important contribution to collaborative intrusion detection research.

Reducing communication overhead is also the motivation in [35]. The use of distributed hash tables in each intrusion detection node is proposed, in order to keep single and correlated alerts. This structure enables efficient and flexible handling of alerts. Computations for each correlated alert are handled to the node with the least load among the nodes relevant to the alert. Moreover, communication issues are also taken into account in the described methodology. Routing of data exchanges between nodes is based on the Kademlia algorithm that ensures that information flow is conducted through the least loaded path. The approach seems interesting, but the actual correlation process is insufficiently analyzed.

The authors in [36] have emphasized on the data fusion part of the alert correlation problem. It is commonly accepted that each intrusion detection sensor is more feasible to detect certain kinds of attacks, according to its nature. The proposed method uses a Neural Network learner unit, that is initially trained with labeled data to decide upon the weights to be used for each intrusion detection sensor. The weights depend on both the sensor itself and the kind of alert it produces. The thresholds used both in each sensor and globally are optimized through the process of observing the data flow and by dynamically modelling normal and anomalous activity distributions. The threshold values are continuously adjusted in order to keep the optimum detection-false alarm trade-off. Experimental results show that the proposed fusion system beneficially combines two different intrusion detection sensors: PHAD and ALAD; the detection rate of the fusion unit is much better than the detection rates of each of the two sensors individually, while the false positive rate is kept at minimum.

5.5. Clustering and hypothesizing on missed events

In [37] the proposed system is a general post-processing solution. Its input is a set of alert sets from multiple intrusion detection sensors. The alerts of each set are aggregated in order to improve their quality, before multiple alert sets merge into one general alert set. Then, a low clustering procedure allows the system to hypothesize about missed security events and to create relevant alerts. The main clustering phase comes next, before the final step, in which a clusters graph is generated to produce a high level presentation of the security events. The system has been tested using the DARPA 2000 dataset, as well as a live network dataset, and has produced satisfactory results.

5.6. Taking into account expert knowledge

Due to the nature of the intrusion detection problem, no automated method can produce a perfect representation of tsecurity state of the protected system. In [38] expert knowledge is used in order to enhance both the intrusion detection and the alert correlation processes. The authors assume that intrusion detection and alert correlation both constitute classification problems. They try to revise results of broadly used classifiers (various Naive Bayes implementations and Decision trees), by taking into account prior expert knowledge. This knowledge is expressed in simple forms, e.g. a certain percentage of traffic is normal or alerts of a certain attack class follow a specific probability distribution. Their algorithm examines this knowledge and tries to alter the results of the classifiers, in order to make them adhere to the relevant limitations, to the extent that this is possible. They finally provide an analytical experimental procedure, using three different datasets, to show the validity of their approach.

5.7. Concentrating on infected hosts

A different approach to the reduction of the size of the alert-set is taken in [39], where the main goal is to find infected hosts. It is difficult to efficiently transform raw alerts to meta-alerts that absolutely correspond to real security events. The authors state that it is easier to just find infected hosts on the protected network, by examining raw alerts and then to further investigate these hosts. They build a novel heuristic to detect infected hosts from a huge alert-set. This heuristic uses a statistical measure to find hosts that exhibit a repeated multi-stage malicious footprint involving specific classes of alerts. Validation of the method showed that it achieves relatively low false positive rates in huge data-sets. It is obvious that the method could be useful to a large network's administrator as she could have a good approximation of infected hosts on her network instead of a very long and impractical alert list.

5.8. Alert flows are more informative than single alerts

In [40] the authors propose that investigating flows of alerts is more effective than investigating single alerts. In this way it is obvious that the size of the data for the analyst;s attention is massively reduced, as flows consisting of alerts related to normal system behavior can contain strong regularities, which can be modeled and eventually filtered out. Normal flow behavior is modeled as a weighted sum of previous observations, using non-stationary autoregressive models. The weights are re-estimated or updated at every new observation. Re-estimation is conducted through the use of a Kalman filter, and it happens on-line, without having to stop examining flows. The most significant differences between forecasts provided by the model and the observations are reported as anomalies and possible intrusions. Finally, these models are used to process voluminous alert flows from an operational network and the results are satisfactory.

5.9. Multiple correlators are better than a single one

The authors in [41] propose a system that is based on multiple correlation methods and, for a given data-set, is able to efficiently combine the results of these methods. A learning phase must exist in advance, in which the performance of each of the correlation methods is measured in terms of their alert reduction rate percentages. The best of the methods are then selected and applied in a best to worst fashion, during the real correlation phase. The experimental process has been conducted on various data-sets while a lot of attention has been given to the total correlation time, as the authors' intention is to produce a system capable of working on-line. An important issue is that the achieved high reduction rate is not an adequate indication of required performance, as the quality of the produced correlated alerts should also be examined.

6. Visualizing results

While all methods previously analysed improve the quality of the produced alert-set, none of them can create an easy to read high level representation for the security analyst. This can only achieved by visualizing the produced alert-set. Despite of this fact, the relevant visualization methods in the literature are not many. The most recent among them are analysed in this section.

6.1. Tables of aggregated alerts

In [42] the motivation is to produce a graphical representation of all possible aggregations of alerts, in order to help security analysts to easily recognize anomalous activity. A graph of tables is created in an hierarchical manner; the root table of the graph represents all events. Each table on the second level represents all possible aggregations produced by defining a specific value for one of the attributes of alerts. The descendants of each second level table are more specific aggregations as the values of a second attribute is picked. It is obvious that nodes on the higher levels of the graph represent more populated aggregations, while nodes on the lower nodes represent more specific aggregations. Probability distributions of attribute values can be useful when searching for anomalies throughout the graph, as a detailed examination of the graph may provide evidence for actual intrusions.

6.2. Limiting the dimensionality of alerts

Usually intrusion detection alerts contain 7-8 interesting attributes. This dimensionality is obviously hard to depict by any visualization method. The motivation of [43] is to research which projection method is the most suitable in order to compress intrusion alert data and make their visualization easier. The methods compared are Principal Component Analysis (PCA), Maximum Likelihood Hebbian Learning (MLHL) and Cooperative Maximum Likelihood Hebbian Learning (CMLHL). The results of relevant experiments have shown that the latter of three methods is the most suitable, as it produces the best results. While this conclusion seems interesting, no indication is given whether CMLHL can be efficiently used in a real world scenario.

6.3. Different views for different uses

An interesting multi-view approach for intrusion detection visualization is presented in [44]. The authors have implemented four different representations, each being suitable for a different scenario. Specifically, there is a main system component responsible for preprocessing and aggregation of alerts along with a PostgreSQL database that holds all required data. The four different views are :

- Daily Summary: A customizable overview which shows various daily summary data, such as aggregated flows per minute over the entire network, or over certain ports.
- Intrusion Detection View: A view based on predefined or user-created templates that shows all relevant intrusion detection alerts. The user

can set criteria, such as port or IP address, for the alerts being shown, in order to see the part of alert flow she is interested in.

- Home centric flow visualization: It consists of a Tree Map that shows traffic flows between attacking hosts and protected network hosts as splines. The size of the TreeMap rectangles (weight), their background color, and the spline width can be set to a default value or they can be computed by some function of the attributes of aggregated flow data, e.g., log of flow count, transferred packets, or bytes.
- Graph-based flow visualization: This is provided as an alternative to the home-centric flow visualization. The main advantage of the graph view is that it emphasizes on the structural properties of the intrusions such as the connectivity between hosts. It is easier to recognize hosts with an intense participation in the intrusion activity.

The last two of the views seem more interesting, but they are not the appropriate views to provide a satisfactory representation of the overall security state of the protected system.

6.4. 3D may be better than 2D

In [45] an innovative approach to intrusion detection visualization is proposed. A 3D graphics engine is used to depict the protected network and the security events. Usually, the means used to visualize intrusion detection data are charts, pies or graphs. In this case, a 3D world is created, in which objects, like hosts or network connections, exist and graphical effects indicate the occurrence of an intrusion. The work presented is in its early stages; not enough evidence exists for the validity of the method.

6.5. Place everything on wheels

An impressive application of radial visualization in the intrusion detection field is presented in [46]. The authors have implemented AlertWheel, which is an intrusion alerts visualization method, based on the bipartite graphs approach. They depict alerts as edges that connect nodes, representative of source IPs, to a central pie, slices of which represent intrusion categories. the number of possible categories does not exceed thirty, while source IPs can be easily grouped in sub-nets. The edges, which correspond to alerts, usually come in huge numbers. The method tries to group these edges whenever they share the same path, in order to produce a readable graph. The security analyst can set criteria to restrict the alerts shown, in order to be able to read the resulting graph more easily. The method is interesting and produces a nice result. Perhaps more alert attributes can be taken into account in order to create a more informative picture.

6.6. A live compact representation

In [47] the author incorporates a three-dimensional graph to depict results of post-processing of intrusion detection alerts. The post-processing method produces clusters of related alerts. It also calculates a validity estimate and a danger estimate for these clusters. A visualization procedure is executed periodically and produces consecutive depictions, which are mainly the frames of the final live representation. In each frame, the clusters are depicted as peaks on an otherwise flat plane. From the graph, the analyst can easily deduce details for each cluster, such as the time range it occurs in; the IPs of the protected network it is related to; the danger estimate for it; and the truth estimate for it. The main objective of this method is to immediately inform the analyst about occurring intrusions in a concise way and enable her to react in time.

7. Discussion

It is obvious from the previous sections that a lot of research has been focused on improving the quality of intrusion detection alert-sets. Through the last decade many methods and approaches have been proposed in order to either reduce false positives, correlate alerts or visualize the outcome in a more elaborate way.

A problem that generally exists in intrusion detection research concerns the data-sets used to test developed methods. The most commonly used data-set is the DARPA data-set [48], which has been heavily criticised as not general enough and out of date. [49],[50],[51]. Some non public data-sets do exist, but they are not used by most researchers, who opt for creating their own, by monitoring intrusion detection alerts in a network they have access to. This makes the comparison among different methods very difficult. A modern data-set, acceptable by all researchers and used by them would help intrusion detection research a lot [52],[53].

While different parts of the problem have been efficiently solved, a complete solution that will generate an outcome free of the usual deficiencies and ready to be read by the security analyst is still missing. First, an efficient combination of methods solving the individual problems mentioned in Section 2 should be chosen. The required interoperability between the methods is an important issue and research should also focus on that. Intensive experiments with various data-sets should be carried out, to reveal possible problems.

Another important issue that needs to be discussed is the flexibility of such a system, as adapting to new kinds of attacks is a necessity. Many methods do perform well, but only after an initial training with a labeled data-set. Training should be repeated periodically each time with an up to date data-set. Only in this way the system will be able to maintain its initial performance. Additionally, new kinds of attacks may prove the system unable to adapt to. From this point of view, methods that do not require training are more appropriate, as long as their performance is satisfactory.

Almost all methods reported herein work on the resulting alert-set and try to alter it, in order to enhance its quality. A different approach to the problem would be to check the alert-set and accordingly modify the configuration of the intrusion detection system, in order to improve its future alerts. Instead of trying to discard false positives or to aggregate identical alerts, it would be wiser to try to expel them from the alert-set in the first place. This would require different implementations for different intrusion detection systems and it would partially solve the problem, as all deficiencies cannot be treated in this way. The combination of this approach with the post-processing methods described herein could produce interesting results.

Post-processing of alerts can improve information about intrusions that have been detected in the first place by the intrusion detection system. It cannot, however, inform the analyst about an event that has not been detected at all. Generally, intrusions consist of sequential actions that correlate to each other. If one of these actions is not detected by the intrusion detection system, the missing information could be reconstructed by studying the detected actions. There are not many research efforts building upon this idea.

While a lot of work has been done on the post-processing of the alerts, there are still many open issues or ideas to be explored for future research. Every proposed method should be adaptable to future attacks; able to cooperate with other methods, in order to be eligible for being part of a complete solution; and general, in order to be efficient, regardless of the intrusion detection systems being used or the systems being protected.

References

[1] A. Valdes and K. Skinner, "Probabilistic alert correlation," in *Recent Advances in Intrusion Detection (RAID 2001)*, ser. Lecture Notes in Computer Science, no. 2212. Springer-Verlag, 2001.

[2] H. Ren, N. Stakhanova, and A. Ghorbani, "An online adaptive approach to alert correlation," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, ser. Lecture Notes in Computer Science, C. Kreibich and M. Jahnke, Eds. Springer Berlin Heidelberg, 2010, vol. 6201, pp. 153–172.

[3] K. Julisch, "Clustering intrusion detection alarms to support root cause analysis," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 4, pp. 443–471, Nov. 2003.

[4] S. Lee, B. Chung, H. Kim, Y. Lee, C. Park, and H. Yoon, "Real-time analysis of intrusion detection alerts via correlation," *Computers & Security*, vol. 25, no. 3, pp. 169 – 183, 2006.

[5] H. Debar and A. Wespi, "Aggregation and correlation of intrusion-detection alerts," in *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, ser. RAID '00, 2001, pp. 85–103.

[6] P. Ning, Y. Cui, and D. S. Reeves, "Constructing attack scenarios through correlation of intrusion alerts," in *Proceedings of the 9th ACM conference on Computer and communications security*, ser. CCS '02. ACM, 2002, pp. 245–254.

[7] X. Qin and W. Lee, "Attack plan recognition and prediction using causal networks," in *Computer Security Applications Conference, 2004. 20th Annual*, 2004, pp. 370–379.

[8] S. J. Yang, A. Stotz, J. Holsopple, M. Sudit, and M. Kuhl, "High level information fusion for tracking and projection of multistage cyber attacks," *Information Fusion*, vol. 10, no. 1, pp. 107 – 121, 2009.

[9] P. Liu, W. Zang, and M. Yu, "Incentive-based modeling and inference of attacker intent, objectives, and strategies," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 1, pp. 78–118, Feb. 2005.

[10] F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer, "A comprehensive approach to intrusion detection alert correlation," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 146–169, 2004.

[11] J. O. Nehinbe, "Automated method for reducing false positives," in *Intelligent Systems, Modelling and Simulation (ISMS), 2010 International Conference on*, January 2010, pp. 54–59.

[12] ——, "Concurrent reduction of false positives and redundant alerts," in *Information Society (i-Society), 2010 International Conference on*, June 2010, pp. 318–323.

[13] G. Victor, M. Rao, and V. Venkaiah, "A bayesian classification on asset vulnerability for real time reduction of false positives in ids," *International Journal of Network Security and Its Applications (IJNSA)*, vol. 4, no. 2, pp. 63–73, March 2012.

[14] H.-S. Lin, H.-K. Pao, C.-H. Mao, H.-M. Lee, T. Chen, and Y.-J. Lee, "Adaptive alarm filtering by causal correlation consideration in intrusion detection," in *New Advances in Intelligent Decision Technologies*, ser. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2009, vol. 199, pp. 437–447.

[15] A. Thomas, "Rapid: Reputation based approach for improving intrusion detection effectiveness," in *Information Assurance and Security (IAS), 2010 Sixth International Conference on*, August 2010, pp. 118 –124.

[16] G. P. Spathoulas and S. K. Katsikas, "Reducing false positives in intrusion detection systems," *Computers & Security*, vol. 29, no. 1, pp. 35 – 44, 2010.

[17] Y. Meng and L.-f. Kwok, "Adaptive false alarm filter using machine learning in intrusion detection," in *Practical Applications of Intelligent Systems*, ser. Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg, 2012, vol. 124, pp. 573–584.

[18] S. Khanchi and F. Adibnia, "False alert reduction on network-based intrusion detection systems by means of feature frequencies," in *Advances in Computing, Control, Telecommunication Technologies, 2009. ACT '09. International Conference on*, December 2009, pp. 513 –516.

[19] J. Treinen and R. Thurimella, "Finding the needle: Suppression of false alarms in large intrusion detection data sets," in *Computational Science and Engineering, 2009. CSE '09. International Conference on*, vol. 2, August 2009, pp. 237 –244.

[20] S. O. Al-Mamory and H. Zhang, "Intrusion detection alarms reduction using root cause analysis and clustering," *Computer Communications*, vol. 32, no. 2, pp. 419 – 430, 2009.

[21] T. Alapaholuoma and J. Nieminen, "A behavior-based method for rationalizing the amount of ids alert data," *ICCGI 2012, The Seventh International Multi-Conference on Computing in the Global Information Technology*, June 2012.

[22] S. Kim, W. Cheng, S. Guo, L. Luan, D. Rosu, and A. Bose, "Polygraph: system for dynamic reduction of false alerts in large-scale it service delivery environments," in *Proceedings of the 2011 USENIX conference on USENIX annual technical conference*, ser. USENIXATC'11. USENIX Association, 2011.

[23] F. Maggi, M. Matteucci, and S. Zanero, "Reducing false positives in anomaly detectors through fuzzy alert aggregation," *Information Fusion*, vol. 10, no. 4, pp. 300–311, October 2009.

[24] G. P. Spathoulas and S. K. Katsikas, "Using a fuzzy inference system to reduce false positives in intrusion detection," in *Systems, Signals and Image Processing, 2009. IWSSIP 2009. 16th International Conference on*. IEEE, 2009, pp. 1–4.

[25] N. Hubballi, S. Biswas, and S. Nandi, "Network specific false alarm reduction in intrusion detection system," *Security and Communication Networks*, vol. 4, no. 11, pp. 1339–1349, November 2011.

[26] A. Mohamed, N. Idris, and B. Shanmugum, "Alert correlation using a novel clustering approach," in *Communication Systems and Network Technologies (CSNT), 2012 International Conference on*, May 2012, pp. 720 –725.

[27] D. Man, W. Yang, W. Wang, and S. Xuan, "An alert aggregation algorithm based on iterative self-organization," *Procedia Engineering*, vol. 29, no. 0, pp. 3033 – 3038, 2012.

[28] A. Kumar, A. Vivekanand, K. Kavitha, M. Gracevennice, and T. Manohar, "Data stream intrusion alert aggregation for generative data stream modelling," *International Journal of Advanced Research in Computer Engineering & Technology(IJARCET)*, vol. 1, no. 7, 2012.

[29] G. C. Tjhai, S. M. Furnell, M. Papadaki, and N. L. Clarke, "A preliminary two-stage alarm correlation and filtering system using som neural network and k-means algorithm," *Computers & Security*, vol. 29, no. 6, pp. 712 – 723, 2010.

[30] R. Vaarandi and K. Podins, "Network ids alert classification with frequent itemset mining and data clustering," in *Network and Service Management (CNSM), 2010 International Conference on*, October 2010, pp. 451 –456.

[31] C. Fung, Q. Zhu, R. Boutaba, and T. Basar, "Bayesian decision aggregation in collaborative intrusion detection networks," in *Network Operations and Management Symposium (NOMS), 2010 IEEE*, April 2010, pp. 349 –356.

[32] M. Ficco and L. Romano, "A correlation approach to intrusion detection," in *Mobile Lightweight Wireless Systems*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2010, vol. 45, pp. 203–215.

[33] B. Morin, L. Me, H. Debar, and M. Ducasse, "A logic-based model to support alert correlation in intrusion detection," *Information Fusion*, vol. 10, no. 4, pp. 285 – 299, 2009.

[34] C. V. Zhou, C. Leckie, and S. Karunasekera, "Decentralized multi-dimensional alert correlation for collaborative intrusion detection," *Journal of Network and Computer Applications*, vol. 32, no. 5, pp. 1106 – 1123, 2009.

[35] Z. Czirkos, M. Rencz, and G. Hosszu, "Improving attack aggregation methods using distributed hash tables," in *ICIMP 2012, The Seventh International Conference on Internet Monitoring and Protection*, May 2012, pp. 82–87.

[36] C. Thomas and N. Balakrishnan, "Improvement in intrusion detection with advances in sensor fusion," *Information Forensics and Security, IEEE Transactions on*, vol. 4, no. 3, pp. 542 –551, September 2009.

[37] G. P. Spathoulas and S. K. Katsikas, "Enhancing ids performance through comprehensive alert post-processing," *Computers & Security*, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404813000503

[38] S. Benferhat, A. Boudjelida, K. Tabia, and H. Drias, "An intrusion detection and alert correlation approach based on revising probabilistic classifiers using expert knowledge," *Applied Intelligence*, pp. 1–21, 2012.

[39] E. Raftopoulos and X. Dimitropoulos, "Detecting, validating and characterizing computer infections in the wild," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, ser. IMC '11. ACM, 2011, pp. 29–44.

[40] J. Viinikka, H. Debar, L. Me, A. Lehikoinen, and M. Tarvainen, "Processing intrusion detection alert aggregates with time series modeling," *Information Fusion*, vol. 10, no. 4, pp. 312 – 324, 2009, ¡ce:title¿Special Issue on Information Fusion in Computer Security¡/ce:title¿.

[41] A. E. Taha, I. A. Ghaffar, A. M. Bahaa Eldin, and H. M. K. Mahdi, "Agent based correlation model for intrusion detection alerts," in *Intelligence and Security Informatics (ISI), 2010 IEEE International Conference on*, May 2010, pp. 89 –94.

[42] J. R. G. J. L. Bogdan Denny Czejdo, Erik M. Ferragut, "Network intrusion detection and visualization using aggregations in a cyber security data warehouse," *International Journal of Communications, Network and System Sciences*, vol. 5, no. 9, pp. 593–602, September 2012.

[43] U. Zurutuza, E. Ezpeleta, l. Herrero, and E. Corchado, "Visualization of misuse-based intrusion detection: Application to honeynet data," in *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*, ser. Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg, 2011, vol. 87, pp. 561–570.

[44] F. Mansmann, F. Fischer, D. A. Keim, and S. C. North, "Visual support for analyzing network traffic and intrusion detection events using treemap and graph representations," in *Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology*, ser. CHiMiT '09. ACM, 2009, pp. 3:19–3:28.

[45] Nurbol, H. Xu, H. Yang, F.-E. Meng, and L. Hu, "A real-time intrusion detection security visualization framework based on planner-scheduler," in *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on*, December 2009, pp. 784 –788.

[46] M. Dumas, J.-M. Robert, and M. McGuffin, "Alertwheel: radial bipartite graph visualization applied to intrusion detection system alerts," *Network, IEEE*, vol. 26, no. 6, pp. 12 –18, November-December 2012.

[47] G. Spathoulas, "Improving intrusion detection alerts," Ph.D. dissertation, Department of Digital Systems, University of Piraeus, 2013.

[48] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 darpa off-line intrusion detection evaluation," *Comput. Netw.*, vol. 34, no. 4, pp. 579–595, Oct. 2000.

[49] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, 2000.

[50] M. V. Mahoney and P. K. Chan, "An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection," in *In Proceedings of the Sixth International Symposium on Recent Advances in Intrusion Detection*. Springer-Verlag, 2003, pp. 220–237.

[51] S. Terry and B. J. Chow, “An assessment of the darpa ids evaluation dataset using snort,” 2005.

[52] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, “Toward developing a systematic approach to generate benchmark datasets for intrusion detection,” *Computers & Security*, vol. 31, no. 3, pp. 357 – 374, 2012.

[53] A. Sperotto, R. Sadre, D. F. van Vliet, and A. Pras, “A labeled data set for flow-based intrusion detection,” in *Proceedings of the 9th IEEE International Workshop on IP Operations and Management, IPOM 2009, Venice, Italy*, ser. Lecture Notes in Computer Science, vol. 5843. Springer Verlag, October 2009, pp. 39–50.