INTRUSION DETECTION ALARM CORRELATION: A SURVEY

Urko Zurutuza, Roberto Uribeetxeberria
Computer Science Department, Mondragon University
Mondragon, Gipuzkoa, (Spain)
{uzurutuza,ruribeetxeberria}@eps.mondragon.edu

Abstract

It is 17 years since Dorothy Denning proposed the first intrusion detection model. These systems have evolved rapidly from that model to present alarm correlation methods. Up to the moment, researchers have developed Intrusion Detection Systems (IDS) capable of detecting attacks in several environments. A boundlessness of methods for misuse detection as well as anomaly detection has been applied. Many of the technologies proposed are complementary to each other, since for different kind of environments some approaches perform better than others. Alert correlation methods try to cover the problem of the huge amount of both positive alarms as well as false alarms they report. The techniques used in this area aim to help the detectors discern between alarms generated by real attacks and legitimate traffic. Consequently, the amount of false alarms can be reduced easing the work of system administrators in relation to IDSs. Proper alert correlation methods also provide a higher confidence for incorporating these systems into organisations.

Keywords
Computer security, intrusion detection, alarm correlation

1. INTRODUCTION

After several years of research on IDS, the variety of results obtained has made the scientific community conclude that further research is needed to fine tune these systems. Large organisation and companies are already setting up different models of IDS from different vendors. Nevertheless, they provide an unmanageable amount of alarms. Inspecting thousands of alarms per day and sensor [1] is not feasible, specially if 99% of them are false positives [2]. Due to this impracticability, during the last four years research on intrusion detection systems has focused on how to handle alarms. The main objectives of these investigation works are: reduce the amount of false alarms, study the cause of these false positives, create a higher level view or scenario of the attacks, and finally provide a coherent response to attacks understanding the relationship between different alarms.

Correlation can be understood as the reciprocal relationship between two or more objects or series of objects. Against the background of IDS, a correlation process can be defined as the necessary steps to be taken in order to discover the connection between different series of security events. The following figure describes the correlation process in accordance with the literature reviewed to write this paper.

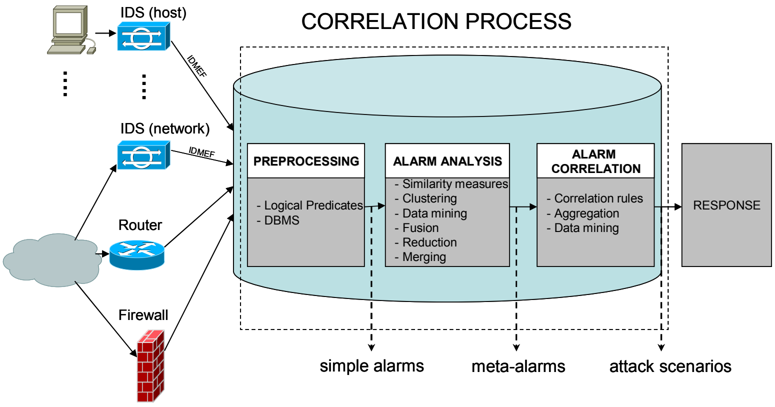


Figure 1. The correlation process

The rest of the paper consists on a survey of the different research works accomplished in the area of correlation applied to IDS. Due to space limitations, the survey is mainly, but not only, focused in the work of Cuppens et al. [3], Ning et al. [4] and Julisch [2]. They have a written a wide range of publications about this subject.

2. THE INTRUSION DETECTION MESSAGE EXCHANGE FORMAT (IDMEF)

When analysing the alarms reported, one of the first problems to solve is the diversity of formats used by different vendor products. Therefore, to be capable of correlate alarms, it is necessary to adapt or pre-process the messages reported by sensors to common format.

Debar and Wespi made it clear in his work about correlation [5] that a unified framework for IDS alerts was essential to handle them independently of the source. The IDMEF developed by the Intrusion Detection Working Group (IDWG) was based in this work. The IDMEF is intended to be a standard data format that automated IDS can use to report alerts about events that they believe suspicious. Nowadays, the IDMEF exists as an Internet Draft waiting for being approved as a RFC. The purpose of the IDMEF is to define data formats and exchange procedures for sharing information of interest to intrusion detection and response systems, and to the management systems that may need to interact with them [6]. The alert data sent to intrusion detection managers by intrusion detection analysers can therefore be represented by the IDMEF data model. IDMEF is an object-oriented representation and an UML model can be found in the Internet Draft, which can be summarized as:

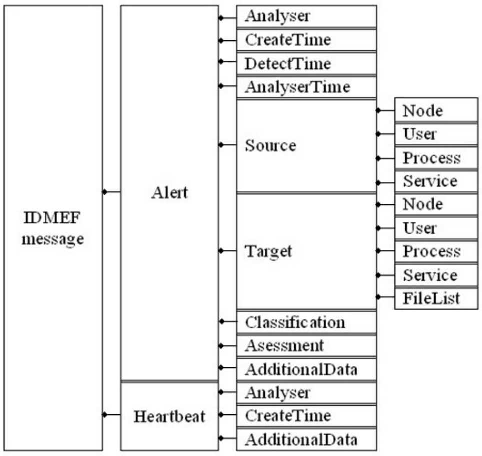


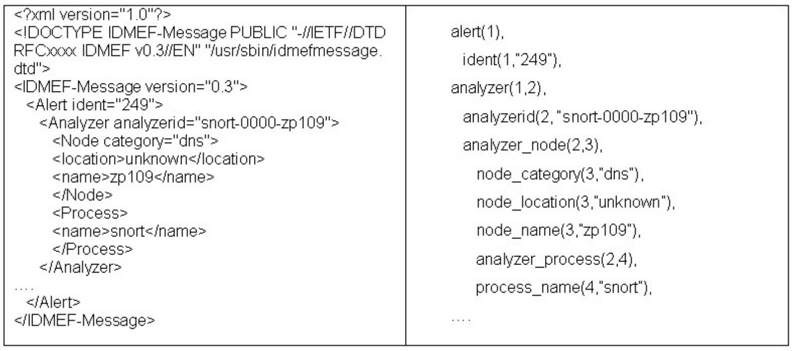
Figure 2. The IDMEF data model

The IDMEF data model has been implemented using a Document Type Definition (DTD) to describe XML documents. In spite of not being a standard yet, the IDS research community has welcomed the use of this format.

3. PREPROCESSING OF ALARMS

Cuppens et al. assume that the format of reported alerts sent by the different sensors will be compliant with the IDMEF format. Then, the XML document will be automatically translated into a set of facts and logical predicates as shown in Figure 3.

After analysing these facts, they are converted into relationships to build a relational data base schema. This way, every XML alert message will be converted into a set of tuples that will be instances of the relational data base schema.



```
<?xml version="1.0"?>
<!DOCTYPE IDMEF-Message PUBLIC "-//IETF//DTD
RFCxxxx IDMEF v0.3//EN" "/usr/sbin/idmefmessage.
dtd">
<IDMEF-Message version="0.3">
  <Alert ident="249">
    <Analyzer analyzerid="snort-0000-zp109">
      <Node category="dns">
      <location>unknown</location>
      <name>zp109</name>
      </Node>
      <Process>
      <name>snort</name>
      </Process>
    </Analyzer>
....
  </Alert>
</IDMEF-Message>
```

```
alert(1),
  ident(1,"249"),
analyzer(1,2),
  analyzerid(2, "snort-0000-zp109"),
  analyzer_node(2,3),
    node_category(3,"dns"),
    node_location(3,"unknown"),
    node_name(3,"zp109"),
    analyzer_process(2,4),
    process_name(4,"snort"),
....
```

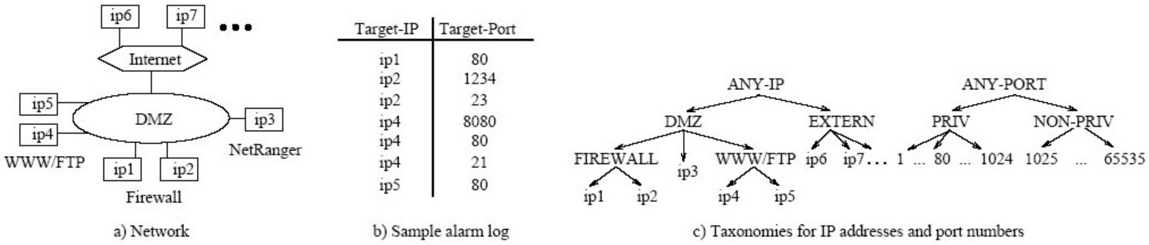
Figure 3. Translation from a IDMEF alarm to a set of predicates

Ning et al. also use logical predicates but they employ them to model the alarms as prerequisites and consequences of attacks. They introduce the idea of "hyper-alert" to represent them for every type of attack. An hyper-alert consists on a triplet of (*fact, prerequisite, consequence*) where *fact* is a set of names of the attributes (each one with its associated values). P*rerequisite* and *consequence* form a logic combination of predicates where its variables can be found in *fact*. Considering a buffer overflow attack against the remote administration tool Sadmind, the type of the hyper-alert will be:

SadmindBufferOverflow= ({VictimIP, VictimPort}, ExistHost (VictimIP) ∧ VulnerableSadmind (VictimIP), {GainRootAccess(VictimIP)})

This method of representation is implemented by a DBMS application so that the knowledge base required for the subsequent correlation is obtained.

Finally, Julich models the alarms as tuples over a Cartesian product (a multidimensional space). These dimension are called alarm attributes. Examples of these attributes can be: source IP address, destination IP address, source port, destination port, type of alarm or timestamp. After that, alarm logs are modelled as a set of alarms. At last, generalization hierarchy or taxonomies are obtained. These are element trees within the domain of some given attributes. The taxonomies are created for every given attribute as shown in Figure 4.



Target-IP	Target-Port
ip1	80
ip2	1234
ip2	23
ip4	8080
ip4	80
ip4	21
ip5	80

a) Network

b) Sample alarm log

c) Taxonomies for IP addresses and port numbers

Figure 4. Network, alarm log, and taxonomies

4. ALARM ANALYSIS

Before the correlation stage, some authors group simple alarms in other higher level ones, thus getting rid of most of the false alarms. Several IDS may send alarms reporting the same event; therefore it is convenient to combine them in the same cluster with attributes of similar characteristics. It will be required to define the technique to best obtain this fusion of simple alarms referring to the same event.

T. Bass introduced data fusion techniques utilized in military applications in order to improve future IDS [7]. The same year, Manganaris et al. apply data mining to alarms generated in multiple sensors of different clients, but their purpose was improving the anomaly detection. The association rules algorithm was applied to discover frequent alarm sets. This way, it is possible to characterize the behaviour of alarms with relation to frequent data sets and association rules [8]. Clifton and Gengo also make use of data mining in order to filter alarms. They look for frequent alarm sequences produced by normal operations. In this manner, they were able to remove most of the false positives. Valdes and Skinner presented the idea of applying probabilistic similarity measures to the fusion of alerts [9] into meta alerts. At the same time, Debar and Wespi implemented a method to aggregate and correlate alarms to show them in more condensed view [10]. An explicit rules algorithm was used to process the alarms that are logically linked. They called them correlation relationships. On one hand, they look for alarms containing identical attributes, called duplicates, which corresponds with the fusion stage. On the other hand they define the consequences; a set of alerts linked in a certain order.

Cuppens et al. employ an expert system where the similarity measures are specified by expert rules. These rules stipulate the similarity relationship between specific attributes of the domain: classification similarity (type of attack), temporal similarity and source and target similarity. After measuring similarity, alert instances are assigned to global alerts (or clusters). A verisimilitude coefficient is given to each cluster. Finally, redundancies are avoided so a specific event emerges just once, even if several alerts have detected it. On one hand, attack source and target are merged and on the other, temporal information.

The description of similarity between alarms given by Julisch is based on defined taxonomies. The closest their attributes are within certain taxonomy, the more similar two alarms will be. Thus, alarms are gathering and they are summarised by a general alarm or cluster. To do so, an Attribute-Oriented Induction data mining heuristic algorithm is implemented. As a result, generalised alarms are obtained and this allows discovering which the root causes of having those alarms are. Removing this causes, Julisch demonstrated that future load of alarms could be reduced over a 90%. He points out that intrusion detection alarms are very homogeneous and repetitive.

A completely different approach is presented by Ning where fusion of alerts is allowed during and after correlation. Continuing with his hyper-alert model, he points out that an alert can be due to a simple alert or to several linked ones. He proposes various utilities in his work for the user of his system to be able to accomplish an interactive analysis of alerts. These utilities are aggregation/disaggregation of alerts, focused analysis, analysis of clustering, frequency analysis, link analysis and association analysis. More information about all these utilities can be found in [4].

5. ALARM CORRELATION

The probabilistic alarm correlator designed by Valdes and Skinner tries to rebuild the attack scenario relaxing the similarity measure of the attack type. This is accomplished for example, designating attributes as similar when, for example, two different alarms are reported with the same source and destination addresses but they have also been reported close in time. Dain and Cunninghan propound an algorithm that generates scenarios by means of the estimation of the probability of a new alert to belong to a certain scenario. To do so, three different approaches to estimate this probability are proposed: naive technique (clustering by IP source address), an heuristic technique (given an scenario, the probability of a new alert to belong to it is resolved considering the most recent previous alert) and data mining techniques. Within data mining, radial base function networks, multilayer perceptrons and decision trees were tested. Decision trees achieved the best results among all these techniques [10].

As mentioned above, Devar and Wespi, carry out the correlation by means of duplicates and consequences. Once executed this correlation, their algorithm performs the aggregation relationship where alerts are added depending particular common characteristics forming "situations", as they call them.

Cuppens et al. automatically create correlation rules declared by means of predicate logic. They first specify logic links between the post-condition of an attack and the pre-condition of another attack. The analysis of attack description defined in the LAMBDA language, which is also based on predicate logic and was developed by them, is used for this purpose. After this, correlation rules are automatically defined offline and rules can be applied. When a new alert arrives, it is checked against the data base, verifying whether another alert is saved in the data base and presents characteristics in the correlation base that show that both, new and saved alert, are potentially correlated. If they are, appropriate rules will be applied to verify whether correlation conditions are fulfilled or not. The result is a set of correlated pairs of alerts where one will be the new alert. Finally, each pair of the set will be examined by an algorithm to see if it should be added to an existing scenario or not. The correlation process of the scenario will be able to generate a scenario alert. Authors present all this in a correlation graph on their interface.

The fact of having attacks specified in LAMBDA language makes it possible for their system to abduct an intermediate alert from a scenario, even if the alarm itself has not been detected, completing the scenario for the attack that took place.

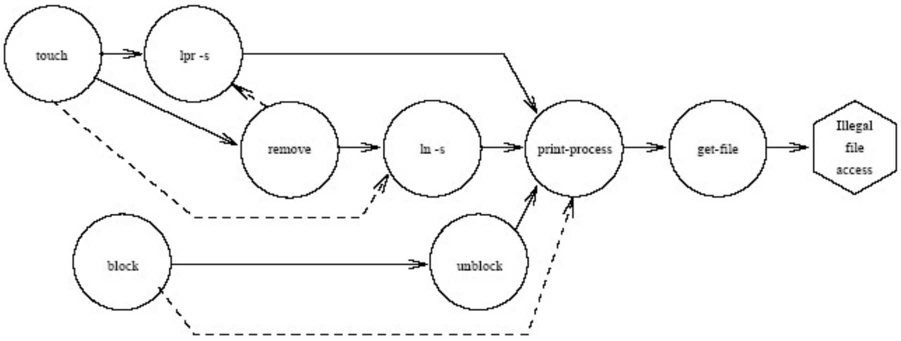


Figure 5. Results of correlation process on attack "illegal NFS mount"

The approach used by Ning is similar to the previous one as it also uses predicates to represent prerequisites and consequences of attacks. To correlate hyper-alerts, whether an hyper-alert contribute to the prerequisite of a later one or not is checked (by its associated interval-based timestamp). The prerequisite is decomposed into predicate parts and they check if the consequence of a previous hyper-alert makes some of the parts of the prerequisite become true. If the result is positive, alerts are correlated. The result is also a graph representing the attack scenario.

6. CONCLUSIONS AND FURTHER WORK

This paper tries to review the main work carried out in the alarm correlation area. Not all authors could be included but we believe that we have chosen the most representative ones. It also necessary to say that not all the authors treat correlation issues the same way. Some of them just correlate alarms from various IDS sensors while others also take into account other important data like network topology, vulnerability assessment tools or firewall and router logs.

Management of alarms and correlation certainly improve the accuracy of IDS significantly decreasing false positives and improving the knowledge of attacks providing a more global view of what is happening by means of the representation of attack scenarios. Despite all the benefits of the techniques presented, they are not perfect and several weak points can be noticed. When some author only correlate alarms taking into account similarities between their attributes, they cannot discover the real reasons of why the alerts are correlated with each other. On the other hand, authors that use previously specified scenarios (by rules or by supervised data mining techniques) to perform the correlation are limited to discover only known scenarios and a great hand work is needed to specify each of them. Other authors propound systems with a great amount of configuration parameters. It is also important that the correlation process is carried out in real time so the response can be more effective.

Another problem found in the literature is that there is not an agreement about the terminology utilized for the different steps of the correlation process. Therefore, some talk about alarm correlation when referring to the clustering and fusion process, while others call correlation to the process of creating of scenarios. We believe that a common criteria is needed.

Finally, the research on IDS is heading towards finding the most suitable response to attacks so they are able to block them before they reach the final stage of the given scenario.

References

[1] Manganaris, S., Christensen, M., Zerkle, D. and Hermiz, K., “A Data Mining Analysis of RTID Alarms”, *Computer Networks 34 (4)*, Elsevier North-Holland, Inc., 2000, pp. 571-577

[2] Julisch, K., ”Clustering Intrusion Detection Alarms to Support Root Cause Analysis”, *ACM Transactions on Information and System Security 6(4)*, ACM Press, 2003, pp. 443-471

[3] Cuppens, F., Miège, A., “Alert Correlation in a Cooperative Intrusion Detection Framework”, *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, IEEE Computer Society, 2002, Berkeley, California, USA, 2002, pp. 202

[4] Ning, P., Peng Ning, Yun Cui, Douglas S. Reeves. “Analyzing Intensive Intrusion Alerts Via Correlation”. *Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID)*, Springer Verlang, Zurich, Switzerland, October 2002, pp. 74-94

[5] Debar, H., and Wespi, A., “Aggregration and Correlation of Intrusion-Detection Alerts”, *Proceedings of the 4th International Symposium on Recent Advances in Intrusion detection (RAID)*, Springer Verlang, California, USA, 2001, pp. 85-103

[6] URL*: http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-12.txt*

[7] Bass, T., “Intrusion detection systems and multisensor data fusion”, *Communications of the ACM*, ACM Press, New York, NY, USA, 2000, pp. 99-105

[8] Clifton, C., Gengo, C., G. “Developing custom intrusion detection filters using data mining”. *2000 Military Communications International Symposium (MILCOM2000)*, Los Angeles, California, USA, 2000, pp. 22-25

[9] Valdés, A., and Skinner, K., “Probabilistic Alert Correlation”, *Proceedings of the 4th International Symposium on Recent Advances in Intrusion detection (RAID)*, Springer Verlang, California, USA, 2001, pp. 54-68

[10] Dain, O. and Cunningham, R.K., “Fusing Heterogeneous Alert Streams into Scenarios”, *In Proceedings of the ACM CCS Workshop on Data Mining for Security Applications*. Barbará and Jajodia, USA, 2001.