Alarm Reduction and Correlation in Defence of IP Networks

Tobias Chyssler[1], Simin Nadjm-Tehrani[1], Stefan Burschka[2], Kalle Burbeck[1]

[1]Dept. of Computer and Information Science
Linköping University, Sweden
[tobch,simin,kalbu]@ida.liu.se

[2] Software & Security Technologies
Swisscom Innovations, Switzerland
Stefan.Burschka@swisscom.com

Abstract

Society's critical infrastructures are increasingly dependent on IP networks. Intrusion detection and tolerance within data networks is therefore imperative for dependability in other domains such as telecommunications or energy distribution. Today's data networks are protected by human operators that are exceedingly overwhelmed by the massive information overload through false alarm rates of the protection mechanisms. This paper studies the role of alarm reduction and correlation in supporting the security administrator in an enterprise network. We present an architecture that incorporates intrusion detection systems as sensors, and provides improved alarm data to the human operator or to automated actuators. Alarm reduction and correlation via static and adaptive filtering, normalisation, and aggregation is demonstrated on the output from three sensors (Snort, Samhain and Syslog) used in a telecom test network.

1. Introduction

The economy and security of modern society is increasingly dependent on a range of Large Complex Critical Infrastructures (LCCIs) such as electricity and telecommunication networks. This paper focuses on the problems – in particular attacks and intrusions – that the operators of an infrastructure management system face upon merger with global information systems that expose them to vulnerabilities of IP networks.

Detection and defence against intrusions in data networks is a field of study in which there are no silver bullets. This is due to the sheer size and diversity of intrusion types. One study [YBU03] estimates that the number of intrusion attempts over the entire Internet is in the order of 25 billion each day and increasing. McHugh [Hug01] claims that the attacks are getting more and more sophisticated while they get more automated and thus the skills needed to launch them are reduced. Most information security (INFOSEC) systems aim to recognise known attacks or anomalies formulated as rules for recognition. Also, there are general tools that simply act as a probe, and facilitate a subsequent use of an IDS, either by data mining on the log results, or by collection of alerts for further examination of a human operator. However, these INFOSEC systems generate far too many alarm reports. While post mortem studies are possible on large data sets, the ability of acting in real-time for counteracting an intrusion is highly dependent on an improved quality of the generated alarms, and in particular reduction of the false alarm rates.

Our study of the range of problems in dealing with security issues in the management network of telecom service providers, has identified the following needs: (1) a general reduction of alarm numbers (2) an improved quality in the produced alarms, i.e. a lower rate of false alarms, (3) means of collating information from many different sources so that unknown attack types and various steps taken by the attacker can be identified via related indices (e.g. combining information about the topology of the network together with IDS alarms), and (4) an indicator of general network "health" with some predictive element so that total service collapse can be avoided. The work in this paper addresses the first two issues and to some extent the third issue that need to be solved before a sophisticated attempt to deal with (3) and (4). Our work provides the first steps to relieve an operator (or an automated agent) from dealing with large volumes of false alarms, in order to concentrate on more intelligent decisions. The work is carried out in the context of the European Safeguard project[Saf03] in which the aim is to demonstrate the use of distributed and coordinated software agents for enhancing existing defence mechanisms in telecom and electricity management networks. This paper does not cover development of new detection or tolerance mechanisms, but contributes towards improving the quality of the data, thus increasing the chance of dealing with serious scenarios by the operator or automated response systems. Being based on well-known INFOSEC mechanisms in IP networks for many infrastructures, we believe that the results can benefit other infrastructures, though here demonstrated on a telecom test network.

The structure of the paper is as follows. First, we present a safeguard architecture that has emerged during the work in the above project. This presents the context of our work. Next, the main contributions of the paper are presented: i.e. how knowledge-based and behaviour-based approaches can be combined to improve the information quality. The knowledge-based approach used by a security expert is used to filter and aggregate alarms, Naïve Bayesian [Hec96] text classification is used to adapt the filters over time, and behaviour-based methods are used to correlate the aggregated alarms. We present exploratory evaluation of three behaviour-based techniques: an additive correlator, a classifier based on Neural networks [Sim98], and a classifier based on K-Nearest-Neighbours (K-NN) [DGL96]. Awaiting publicly available GCP benchmarking data, we have been evaluating our methods on data generated on a test network. The test network consists of more then 50 machines, and has been set up at a major telecom provider (Swisscom).

There exist a number of proposals for alarm correlation (e.g. [CM02], [DW01], [MD03], [NCR02] and [PFV02]). These systems are limited to perform correlation of alarms that match predefined rules for scenarios or attack conditions and consequences. Qin and Lee propose a method to find new scenarios [QL03]. Another approach taken by Valdes and Skinner is the probabilistic alert correlation [VS01], where they use a mathematical framework to fuse alerts based on how well they match each other. Our method for correlation is to our knowledge the first to apply behaviour-based correlation using neural networks and K-nearest neighbours.

The idea to perform data mining of alarms to find filters for false alarms has been explored by Julisch and Dacier [JD02], where they use conceptual clustering of old alarms to derive new filters. Our method for adaptive filtering has the same objective, but we use text classification of the alarms.

2. The Safeguard context

The architecture of the Safeguard agents is presented in Figure 1. The key to a generic solution applicable in many infrastructures is in the definition of *roles* for various agents. These are believed to be common for the defence of many infrastructures, but should be instantiated to more specific roles in each domain.

Wrapper agents – wrap standard INFOSEC devices and existing LCCI diagnosis mechanisms, provide their outputs after some filtering and normalisation for use by other agents.

Topology agents – gather dynamic network topology information, e.g. host types, operating system types, services provided, known vulnerabilities.

Hybrid detector agents – adapted to the domain of a given infrastructure, but combine knowledge and behaviour based mechanisms (e.g. data mining with white lists).

Correlation agents – identify problems by using several sources of information from wrapper, topology, or hybrid detector agents. Use the data sources to order, filter and focus on certain alarms, or predict reduced availability of network critical services.

Action agents – enable automatic and semi automatic responses when the evaluation of a problem is finished.

Negotiation agents – communicate with agents in other LCCIs to request services and pass on information about major security alarms.

HMI (Human-Machine Interface) agents – provide an appropriate interface, including overview, for one or many system operators.

Actuator – wrapper for interacting with lower layer software and hardware (e.g. changing firewall rules).

There may be several instances of each agent in each LCCI, and in particular, we are developing several types of correlation agents, of which only some variants are described in this paper.

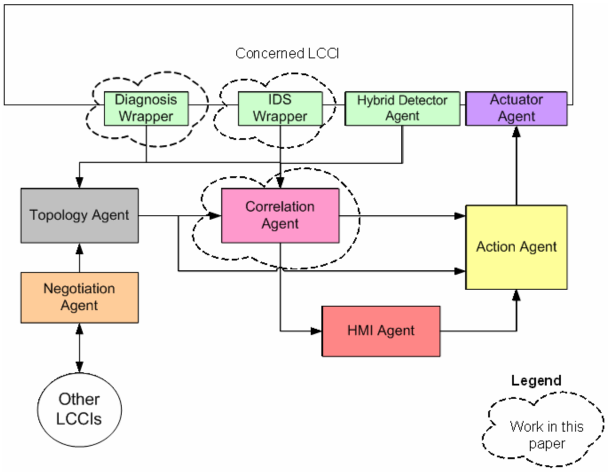

Figure 1: The Safeguard agent architecture

3. Alarm Reduction

In this paper we report on the use of three different INFOSEC devices: One network based system, Snort (rule-based detector that inspects network packets), and two host-based systems, Samhain (a file integrity checker) and Syslog (system log messages). Ongoing work includes more sensors, such as an anomaly detector for network traffic and multiple sources of topology information.

We have studied how a security expert currently monitors and analyse alarms, covering some ideas used in active intrusion detection. The analysis was then adapted to the three INFOSEC devices that are used in this paper. There are many triggers for alarms that should be investigated, for example: alarms with high severity, hosts with a lot of alarms, hosts with a lot of different alarms, unusual events, high rate of alarms, and strange payload in "normal" alarms.

Reviewing how a security expert works, some features can be noted that will be reflected in our agents that mimic the behaviour: (1) Port scans are aggregated so that each port scan only generates one alarm. (2) When analysing the alarms, all three sources are used around the time of interest. (3) Knowledge about the topology of the network is captured to assist the decisions.

Overall, the alarm analysis activities can be broadly grouped into filtering, aggregation and correlation. Using the sum of the severities during a time period captures some of the characteristics for listed alarms (high severity, lots of alarms and high rate).

Figure 2 presents an overview of the implemented operations in our agents based on the above description of activities. Normalisation is the processing needed for getting the data from different sensors into a uniform format.

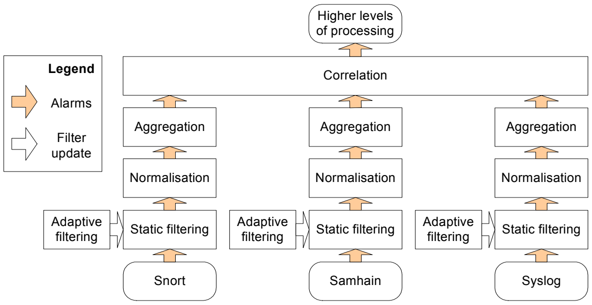

Figure 2: Overview of methods

3.1 Static filtering

Studying the output of the IDSs when no attacks are launched shows that there are a lot of uninteresting alarms coming from Syslog and Samhain. Even though each of them has a low severity, their severity sum is high depending on their vast numbers. For example, each time a file is checked for changes, an alarm is produced. Based on this study filters were inserted to remove the uninteresting alarms, such as messages that Samhain checks a file or that Syslog is doing garbage collecting while idle, filters for messages that arose due to current misconfigurations, and a filter that removed port scan alarms in Snort where the source is the central Syslog and Samhain server.

3.2 Adaptive filtering

Manually studying the output of the IDSs and adding filters can solve the problem with uninteresting and misconfiguration messages for the current set-up. However, since it is not possible to foresee all future misconfigurations, adaptive filtering algorithms were implemented to update the filters. An automatic filter detector based on Naïve Bayesian (NB) learning was designed for Syslog. The idea is to train a NB text classifier to classify messages by looking at the words that appear in messages. The NB-classifier is first trained by presenting a number of messages labelled as interesting or not. The meaning of interesting is a message that can be useful when analysing it for signs of an attack (for examples see section 4.2). During training, a knowledge base is built up by counting the occurrences of the tokens (words) in the message. After training, the on-line classification is based on computing the probability that an object belongs to a class based on how common the values of the attributes (words) are in that class within the training data.

The adaptive filters are used in the following workflow. On a periodic basis or on demand, the algorithm for adaptive filtering is launched. The algorithm's classification is presented to the HMI agent (typically to reach a human expert). In order not to overload the receiver, only the top "scoring" messages are selected (e.g. messages with a high count, which are those that have the most influence). The HMI then replies with its classification. The reply is also used to optimise the training corpus (thus achieving on-line retraining).

3.3 Aggregation

Port scans generate a lot of messages in Snort, and are not of primary interest. It would reduce the information overload if each scan results in only one alarm. The found sources are useful as an additional parameter when looking for other attacks, since it is known that network scans can be the first sign of an attack. Other alarms can also be aggregated if they are similar and are within a certain time from each other. This is true for all IDS. However, which alarms that are treated as similar are specific to each IDS. For example, Snort alarms in which the source IP, the destination IP, and the signature is the same within a given time window are aggregated.

3.4 Correlation

For the correlation that is solely based on the alarm data from the three wrappers (before correlating with topology or other temporal information), three different ways to correlate were explored and compared to each other. The assumption is that when attacks occur, there should be alarm patterns (alarms with severities) present in some INFOSEC mechanism during that time.

All alarms about one host during a time window are considered, i.e. the selected correlation parameters are time and IP address. Given a time window, a time slot of three dimensions is produced by taking the sum of severities from the three sensors. Based on the input pattern, the correlator will try to decide whether there is an attack or not. The idea is illustrated in Figure 3.

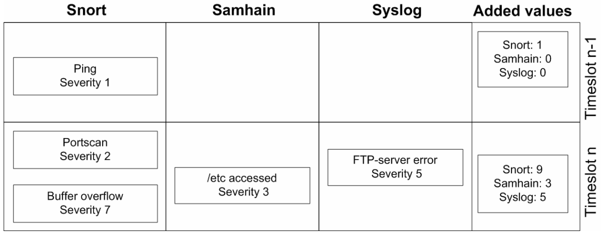

Figure 3: Alarms appearing in two time slots

The three-dimensional vectors were tested as inputs to the following three correlation algorithms: (1) a simple additive correlator with parameter 'severity sum threshold', (2) a neural network with parameters number of neurons, number of hidden layers, weights of the connections and (3) a K-NN classifier with parameter K.

The additive correlator simply adds the three severities. The sum is compared to a threshold, and if it is larger than the threshold it generates an alarm, otherwise not. By varying the threshold different detection rates and false positive rates can be achieved. The other two correlation algorithms are first trained with some time slots and are then used to classify new time slots. More details about these two correlation algorithm parameters can be found in [Chy03] that also describes each technique in more detail.

4. Evaluation

This section presents the evaluation of the presented techniques using the data generated by the test network constructed for this purpose. The topology of the test network used to perform the evaluation is given in [Chy03]. The major parts of the network are a server zone, a workstation zone and an external zone with nodes interconnected by routers and switches. There are Sparc and x86 machines running different types of operating systems including Solaris, Linux and Windows NT. The external zone represents the Internet when attacking the network from outside. Data including attacks and "normal" data for evaluation can be collected from the network and labelled.

In our experiments the attacks were performed using various tools and techniques, including: (1) Scanning of the network using ping script once, (2) Security scan with Nessus[1] four times, (3) security scan with Nmap[2] five times, (4) Brute-force password guessing for telnet with Brutus[3] used twice, (5) Sadmin buffer overflow attack launched against two different hosts at separate times, (6) Installing a root kit[4] for Solaris 2.6 once, (7) Various "bad" behaviour when logged in on a computer (e.g. allocating all space on the disk, killing as many processes as possible).

4.1 Static filtering

The data used to test the result of static filtering is gathered from all relevant hosts on the network during three weeks and includes alarms generated from attacks and from normal traffic. Figure 4 illustrates the result of static filtering on the Syslog, Snort and Samhain alarms. No alarms related to attacks were removed.

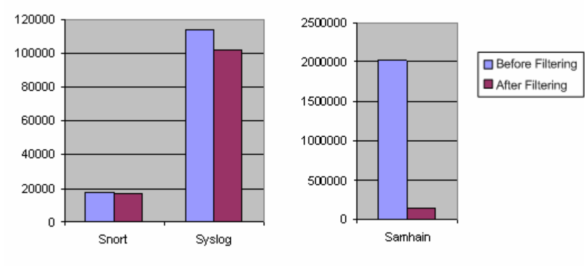

Figure 4: Static filtering for Snort/Syslog/Samhain

Clearly Samhain alarms were reduced most drastically by this knowledge-based approach (order of

[1] www.nessus.org
[2] www.insecure.org/nmap
[3] www.hoobie.net/brutus
[4] www.honeynet.org/papers/motives

magnitude). The method had the lowest impact on Snort alarms. The results can be explained by the fact that Snort is a misuse detection IDS while Syslog and Samhain are not IDSs by definition.

4.2 Adaptive filtering

Next we describe the success of the text classification method for adaptive filtering, illustrated on the Syslog alarms. A data set was specifically generated for this purpose and was labelled by a security expert to be used for training the text classifier. The data set consisted of messages during two months from every host running Syslog in the network. The label is not whether the message is a true alarm or not, rather what is called “interesting” or “uninteresting”. For example, a message saying “telnetd[1]: Successful login as user root” or “File changed” is classified as interesting, but messages like “Syslog-ng: 54 Objects alive. Garbage collecting while idle” or “Could not resolve host name” will be classified as uninteresting (the last message is an example of a misconfiguration that should be detected and reported separated from the intrusion detection process). When classifying the messages, facility, priority, program and the message part of each Syslog message was included. Level was excluded since it has the same value as priority. The date, time and tag were also not used since it was deemed as uninteresting when classifying the messages. The host was excluded since the same message coming from two different hosts is often equally interesting.

The data set contains 156 212 alarms, where 18 941 of them are classified as interesting. These alarms where then divided into one set for training and another set for testing. The test data set contained 53 722 alarms, where 10 621 of them are classified as interesting. The other 102 490 alarms were used for training. Notice that there is a difference in the rate of interesting alarms between the training data set and the test data set (8.1% and 19.8% respectively), something that can influence the results for methods based on statistics such as naïve Bayesian learning. Table 1 shows the results of the adaptive learning algorithm in terms of the precision of the results produced on the test data set.

Data set	Correct	Incorrect	Precision
Test part	53682	40	0,99926

Table 1: Text classification for adaptive filtering

By precision here we mean the number of correct classifications divided by the total number of alarms. As the last column indicates, the method classifies the alarms with a very high precision as long as the selected attributes for classifying the alarms remains valid in the eyes of the security expert.

4.3 Aggregation

Next we show the result of aggregation, here presented for Snort alarms. Before aggregation there were 10 162 alarms, where 4 985 of them were port scan alarms. Tests were performed using variations in the size of the time window. With the shortest time window (0.5 minute), the 10 162 alarms could be reduced to less than 2 500. The method does not reduce the false alarm rates, but can help to keep the alarm rates down in order not flood the receiver. Obviously, the larger the time window the fewer alarms are left after aggregation. However, higher-level network correlation agents are dependent on recognition of anomalous situations within short enough times for reaction [Bur03].

4.4 Correlation

Next we present the comparison of the three explorative techniques for behaviour-based correlation in order to focus on the significant alarms. The three illustrations cover the additive correlator, a neural network and a 1-Nearest Neighbour (1-NN, the best instance of K-NN) classifier algorithm. A data set was constructed for the correlation experiments using all alarms regarding eight hosts on the network during roughly eight days. Using a time window of 15 minutes, 6 048 three-dimensional time slots (as described in 3.4) were created. This data set was labelled by a security expert for the purpose of evaluating the learning-based correlation techniques. 54 of the time slots were part of an attack and thus labelled as attack, the rest as not part of an attack. The data set was then divided into a training part (constituting of 3993 time slots where 36 of them were attacks) and a test part (the rest of the time slots). Figure 5 shows results of the evaluation on the test data.

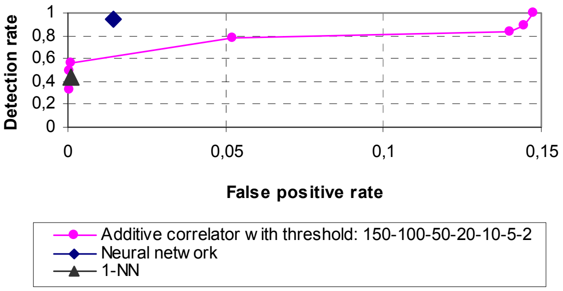

Figure 5: ROC curve for correlation methods.

The additive correlator simply compares the sum of the three values in the time slot with a pre-selected threshold. Obviously a higher threshold gives less false positives (as visible on the ROC curve of Figure 5).

The figure shows that the 1-NN algorithm and the neural network have lower levels of false-positive compared to the additive correlator with a threshold of 20. However, the additive correlator is a simple approach and does not require any training but still produces a better classification result compared to the K-NN approach. The neural network correlator has a superior behaviour but it requires a costly training process.

5. Conclusions

In the above studies three methods that need labelled data during training (naïve Bayesian learning, neural networks and K-NN) and one method that does not need any training (additive correlator) were evaluated. The NB classification was used for generating adaptive filters, and the three other methods were compared as possible techniques for correlation. The drawback of using supervised learning (training with labelled data) is that training is a time-consuming process. Also, much time is spent on collecting and labelling data, and the labelling usually has to be done by an expert in the field. The NB-classifier has the benefit that it can be subjected to on-line retraining based on the presented workflow and interaction with the human expert presented in section 3.2. All three correlators showed high rates of accuracy. However, the method with the highest success was also the one that needed the larger effort in terms of training, and identification of configuration parameters. The non-learning method, as well as being simpler to set up, has the advantage of only having one parameter that can be adjusted to change the performance. This makes it easier to use for operators without knowledge of the underlying techniques; something like a knob that the operator can rotate in different directions to increase or decrease the detection rate at the cost of more false positives.

A full evaluation of the work is dependent on applying the same technique on the GCP or other publicly available data on which other methods are also evaluated. The results in section 4 above illustrate the success of the combined filtering, aggregation and correlation in satisfying the immediate needs of the security experts using the given sensors (i.e. reduction of number of alarms, as well as a reduced set of false alarms). The static filtering is especially valuable when used on non IDS INFOSEC mechanisms. Current work includes the further processing of the data provided by these agents for correlation with anomaly data as well as topology information, and the monitoring of the overall health of the network in terms of provision of its critical services.

6. References

[Bu03] K. Burbeck, S. G. Andres, S. Nadjm-Tehrani, M. Semling, and T. Dagonnier. “Time as a Metric for Defence in Survivable Networks”. Proceedings of the Work in Progress session of 24th IEEE Real-Time Systems Symposium (RTSS 2003), Dec. 2003.

[Chy03] T. Chyssler. “Reducing False Alarm Rates in Intrusion Detection Systems”, Master thesis No. LITH-IDA-EX-03/067-SE, Linköping University (2003).

[CM02] F. Cuppens and A. Miége. “Alert Correlation in a Cooperative Intrusion Detection Framework”. Proceedings of the 2002 IEEE Symposium on Security and Privacy. 2002. Pages 187 – 200.

[DGL96] L. Devroy, L. Györfi and G. Lugosi. “A Probabilistic Theory of pattern Recognition”. Springer Verlag, New York Inc, 1996.

[DW01] H. Debar and A. Wespi. “Aggregation and Correlation of Intrusion-Detection Alerts”. RAID. Springer Verlag, Oct. 2001. Pages 85-103.

[Hec96] D. Heckerman. “A Tutorial on Learning With Bayesian Networks”. Technical Report MSR-TR-95-06, Microsoft Research. March 1995 (Revised November 1996).

[Hug01] J. McHugh. “Intrusion and Intrusion Detection”. International Journal of Information Security. Vol 1, No 1. Springer Verlag, Aug. 2001. Pages 14 – 35.

[JD02] K. Julisch and M. Dacier. “Mining Intrusion Detection Alarms for Actionable Knowledge”. Proc. of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM Press, Jul. 2002.

[MD03] B. Morin and H. Debar. “Correlation of Intrusion Symptoms: an Application of Chronicles”, RAID. Springer Verlag, November 2003. Pages 97 – 112.

[NCR02] P. Ning. Y. Cui and D. S. Reeves. “Constructing Attack Scenarios through Correlation of Intrusion Alerts”. Proc. of the 9th ACM conference on Computer and communications security. ACM Press, 2002. Pages 245 – 254.

[PFV02] P. A. Porras, M. W. Fong and A. Valdes. ”A Mission-Impact-Based Approach to INFOSEC Alarm Correlation”. RAID. Springer Verlag, October 2002. Pages 95–114.

[QL03] X. Qin and W. Lee. “Statistical Causality Analysis of INFOSEC Alert Data”. RAID. Springer Verlag, November 2003. Pages 73 – 93.

[Saf03] Safeguard website: http://www.istsafeguard.org, Acc. May 2004.

[Sim98] J. Sima. “Introductions to Neural Networks”. Technical report No V-755, ICS CAS, Prague, 1998.

[VS01] A. Valdes and K. Skinner. ”Probabilistic Alert Correlation”. RAID. Springer Verlag, October 2001. Pages 54-68.

[YBU03] V. Yegneswaran, P. Barford and J. Ullrich. “Internet Intrusions: Global Characteristics and Prevalence”. Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and Modelling of Computer Systems, June 2003. Pages 138-147.