INTRUSION ALERT QUALITY FRAMEWORK FOR SECURITY FALSE ALERT REDUCTION

NAJWA ABU BAKAR

UNIVERSITI SAINS MALAYSIA

2007

INTRUSION ALERT QUALITY FRAMEWORK FOR SECURITY FALSE ALERT REDUCTION

by

NAJWA ABU BAKAR

Thesis submitted in fulfilment of the requirements for the degree of Master of Science

June 2007

ACKNOWLEDGEMENTS

Praises to Allah Almighty for giving me the strength and courage to complete this research. Thank you to the Ministry of Science, Technology and Innovation (MOSTI) for granting me the National Science Fellowship (NSF) Scholarship; my supervisor, Associate Professor Dr Bahari Belaton for his excellent supervision; Associate Professor Dr Azman Samsudin for co-authoring paper publication; system analysts, En Mahadi and En Norazman for their technical support; the faculty of the School of Computer Sciences in Universiti Sains Malaysia (USM); and all the staffs of Institut Pengajian Siswazah (IPS) for their cooperation.

Thanks to Raja Azrina, Megat, and Adli from National ICT Security & Emergency Response Centre (NISER), Kamal from Defenxis, and Fadzil from Intel for providing the data and supporting me with technical information. Much appreciation to Sis Masitah for editing and proofreading the thesis and to my friends, Nur Hana, Lim Lian Tze, Adib, Hussein and many more who helped me directly and indirectly throughout my research.

Special thanks to my husband for his support, patience, and understanding from beginning to the end, my parents for the endless prayers, and my children for the inspiration.

TABLE OF CONTENTS

CHAPTER FIVE : EXPERIMENTS AND RESULTS

CHAPTER SIX : CONCLUSION AND FUTURE WORK

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

ACC	Aggregation and Correlation Component
AI	Artificial Intelligence
CATCH	Comprehensive Assessment for Tracking Community Health
CERT	Computer Emergency Response Team
CIDF	Common Intrusion Detection Framework
CRIM	Cooperation and Recognition of Malevolent Intentions
CS	Computer Science
CVE	Common Vulnerabilities and Exposures
DARPA	Defense Advanced Research Projects Agency
DDoS	Distributed Denial of Service
DMZ	Demilitarized Zone
DoS	Denial of Service
DTD	Document Type Definition
EMERALD	Event Monitoring Enabling Responses to Anomalous Live Disturbances
HIDS	Host Intrusion Detection System
IAC	Intrusion Alert Correlation
IAQF	Intrusion Alert Quality Framework
ICT	Information Communication Technology
IDMEF	Intrusion Detection Message Exchange Format
IDS	Intrusion Detection System
IPV6	Internet Protocol version 6
IQ	Information Quality
ISN	Initial Sequence Number
LAN	Local Area Network
LLDOS	Lincoln Lab Denial of Service
MIM	Man-in-the-Middle
MIT	Massachusetts Institute of Technology
NIDS	Network Intrusion Detection System
OS	Operating System
P0f	Passive OS Fingerprinting
PADS	Passive Assets Detection System
PVS	Passive Vulnerability Scanner
QC	Quality Classifier
RAM	Random Access Memory
RNA	Real-time Network Awareness
SIG	System/Network Information Gathering
SIM	Security Information Management
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
TDQM	Total Data Quality Management
TIAA	Toolkit for Intrusion Alert Analysis
TQM	Total Quality Management
USM	Universiti Sains Malaysia
VKB	Vulnerability Knowledgebase
VPN	Virtual Private Network
WWW	World Wide Web
XML	Extensible Markup Language
XSS	Cross Site Scripting
XST	Cross Site Tracing

PENGURANGAN AMARAN KESELAMATAN PALSU MENGGUNAKAN RANGKA-KERJA KUALITI AMARAN PENCEROBOHAN

ABSTRAK

Tesis ini mengkaji rekabentuk dan perlaksanaan rangka-kerja yang mempersiapkan amaran-amaran keselamatan dengan metrik-metrik yang disahkan, memperkayakan amaran-amaran keselamatan dengan metrik-metrik tersebut dan akhirnya, menyeragamkan amaran-amaran tersebut dengan satu format yang dipersetujui agar sesuai digunakan oleh prosedur-prosedur penganalisaan amaran di peringkat tinggi. Rangka-kerja ini dinamakan “Rangka-kerja Kualiti Amaran Pencerobohan” (IAQF) dan tujuan utamanya adalah untuk menambahbaik pengurangan amaran keselamatan palsu dalam bidang pengesanan pencerobohan. Satu analisa ke atas penyelesaian-penyelesaian sedia ada untuk mengurangkan amaran-amaran palsu menunjukkan yang mereka tertumpu sama ada pada peringkat penggera atau peringkat penganalisaan. Merubah atau menyesuai penggera-penggera mungkin membantu mengurangkan bilangan amaran-amaran tetapi kita berisiko untuk terlepas pandang serangan-serangan sebenar yang dikenali sebagai negatif yang palsu. Pada peringkat yang lain, menyerahkan tugasan untuk menapis amaran-amaran palsu pada peringkat penganalisaan mungkin tidak juga berkesan. Pertama, kerana informasi yang berkaitan amaran-amaran tersebut tidak lengkap membuatkan sebarang pilihan efektif pada peringkat ini sukar dan hasilnya pada kebiasaannya tidak tepat. Kedua, amaran-amaran dalam jumlah yang amat besar mungkin mendominasi masa pengiraan untuk memperbaiki amaran-amaran tersebut sebelum sistem dapat melaksanakan tugasan-tugasan utama iaitu pengurangan amaran-amaran palsu. Oleh itu, persiapan data yang teratur perlu sebelum amaran-amaran dianalisa. Dalam kajian ini, kami melihat masalah ini dari perspektif pengurusan informasi di mana masalah ini adalah disebabkan oleh kualiti data amaran-amaran yang rendah. IAQF yang mengadaptasi prinsip kualiti data yang dipanggil TDQM dicadangkan di mana proses-prosesnya

terdiri dari penakrifan, pengukuran, penganalisaan, dan penambahbaikan. IAQF telah dilaksanakan pada peringkat awal prosedur-prosedur analisis amaran untuk mempersiapkan dan menambahbaik kualiti data amaran-amaran tersebut. IAQF adalah bercirikan kebolehan untuk mengesahkan amaran-amaran menggunakan sumber informasi yang berkaitan, memperkayakan amaran-amaran dengan metrik-metrik kualiti data, dan menyeragamkan amaran-amaran menggunakan format IDMEF, satu format data yang dipersetujui untuk mempersembahkan amaran-amaran. Kelebihan pendekatan ini adalah hasilnya yang boleh terus digunakan oleh prosedur-prosedur penganalisaan iaitu perkaitan, perlombongan data, dan mesin pembelajaran. Kami telah menunjukkan bahawa dengan melaksanakan prinsip kualiti data terhadap pengurangan amaran palsu, kami telah berjaya mengurangkan amaran-amaran palsu antara 10 hingga 50%, dan mempersiapkan amaran-amaran dengan informasi berkaitan tambahan untuk kebaikan penganalisaan di peringkat tinggi.

INTRUSION ALERT QUALITY FRAMEWORK FOR SECURITY FALSE ALERT REDUCTION

ABSTRACT

This thesis investigates the design and implementation of a framework to prepare security alerts with verified data quality metrics, enrich alerts with these metrics and finally, format the alerts in a standard format, suitable for consumption by high-level alert analysis procedures. This framework is called “Intrusion Alert Quality Framework” (IAQF) and its main aim is to improve false alerts’ reduction in intrusion detection area. An analysis of existing solutions to reduce false alerts shows that they focus either at the sensor-level or at the analysis-level. Tuning or customizing the sensors may help reduces the number of alerts but we risk missing real attacks known as false negative. On the other extreme, leaving the tasks to filter false alerts at the analysis stage may not be effective either. First, is because incomplete contextual information about alerts may make any effective decision at this stage difficult, and the outcome to be most likely inaccurate. Second, the sheer size of alerts may dominate the computational time of cleaning raw alerts prior to performing the core task of reducing false alerts. Thus, a proper data preparation at low-level stage nearer to the data source is needed prior to the alert analysis. In this research, we look at this problem from the information management perspective where the problem is due to the alerts’ low data quality. IAQF that adapts a data quality principle called TDQM is proposed where the processes included are definition, measurement, analysis, and improvement. IAQF is implemented at the low level stage of alert analysis procedures to prepare and improve the data quality of the alerts. IAQF features the ability to verify alerts using resource contextual information, enrich them with data quality metrics, and standardize them using IDMEF format, a standard data format to present IDS alerts. The advantage of this approach is that the output can be directly consumed by analysis procedures, which are correlation, data mining, and machine learning. We

demonstrated that by applying data quality principles towards false alerts reduction, we managed to reduce false alerts in the range of 10 to 50%, and prepared the alerts with extra contextual information to benefit the high level analysis.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Data quality management has been practiced to solve *low data quality* problem in information systems of organizations such as business (English, 1999), healthcare (Berndt et al., 2001), and WWW (Naumann, 2002). This problem exists most of the time in information systems as a result of incorrect entered, incompletely stored, inaccurately produced, or outdated data. This issue is severe especially if the data are very large in volume, as spotting and filtering inaccurate data use a lot of resources and are usually costly. Hence, a proper data quality management is needed to control the quality of the produced and stored data. Deployment of data quality management at strategic level, such as during the initial stage of data production may reduce costs and the resources needed to process data at the high-level stage, such as data analysis and data mining (Pyle, 1999).

Low data quality scenario also affects Information Communication Technology (ICT) security field, specifically in the intrusion detection area. Security analysts in typical organizations that implement ICT security have been bombarded with huge amount of *security alerts* or logs, generated by security sensors, like Intrusion Detection System (IDS), firewalls, and routers. In the context of this thesis, security alert is an alarm, a log, or a warning report, triggered by signature-based IDS sensors when an attack signature is identified. The generated IDS alerts, not only normally are huge in volume, but also lack contextual information, and contain low quality data. Therefore, in this research, we studied the effect of applying data quality management towards these alerts. The immediate result of this research is to improve the low data quality issue in IDS, by

producing high data quality alerts as input for the high-level analysis, so that false alerts are reduced.

This chapter reviews the low data quality alert in IDS, and proposes solutions for the problem. In Section 1.2, the problem statement is presented. In this section, we detail out what factors contribute to low data quality alert as well as the problems with the current solutions. Section 1.3 proposes the solution, states the research objectives, and provides the scope of the work. Then, Section 1.4 lists out the terminologies while Section 1.5 details out the main contributions. Finally, Section 1.6 describes the thesis outline.

1.2 Problem Statement: False Alerts in Intrusion Detection

Although the purpose of an alert is to warn security analysts when intrusion occurs, sensors have their own weakness that is producing a lot of false alerts such as false positives and noises. This is IDS's traditional problem. According to Northcutt et al. (2003), false alerts are triggered as a result of sensors' signature rules that are too general. Signatures are purposely not constructed with rules that are too specific to avoid *false negative*. Rules that are too specific will miss potentially malicious packets because attacks launched by intruders are constantly evolved from time to time to avoid detection. This attack development can be done by crafting the packets. However, signatures that are too general cause sensors to accept all packets that match the signatures and produce alerts that might be false. Because of this, sensors generate thousands of alerts that are benign where destination hosts are not vulnerable to the attack. Knowing this drawback, intruders may craft the packets that purposely launch fake attacks that trigger large amount of benign alerts and paralyze sensors from detecting real attacks. To make the condition worse, this scenario is directly proportionate to the number of sensors deployed in a network. Security analysts' attention is often diverted to respond to thousands of

generated false alerts with low possibility to succeed while true alerts take a long time to be identified, or worse, might just be ignored. This is because sifting through these alerts to extract the true ones is a very time consuming task. This false alert problem deteriorates the IDS basic function that is to detect intrusions and immediately warn security analysts by producing alerts.

Beside this phenomenon, another problem that contributes to low data quality alerts is IDS sensor outputs that are not standardized. Different types of sensors produce alerts in various forms that make alert analysis difficult. Most organizations apply defense-in-depth strategy where more than one security devices like IDS and firewall, are deployed in the network. Some organizations might also install more than one type of IDS sensors such as host-based, network-based, signature-based, or anomaly-based IDS. The purpose of deploying different types of sensors is to rely on other security devices in case one of them fails to produce any alert. In order to benefit from this defense-in-depth practice, the outputs from these devices need to be correlated. Thus, standardization is needed to increase data quality of the alerts to help the correlation process.

Currently, a lot of high-level alert analysis systems developed, have been directly or indirectly solving the IDS false alert problem discussed before by implementing *analysis methods* such as aggregation, correlation, data mining, and machine learning. At this level, the analysis methods are implemented mainly for the purpose of identifying true alerts, understanding attack scenario, or making critical security decisions. These sophisticated analysis methods have been shown to successfully identify true alerts, and at the same time marginally reduce false alerts. However, the systems that implemented these methods take the input alerts directly from the IDS sensor's database or log file. As mentioned before, these raw alerts data quality is low, since the accuracy of the alerts is

unknown, the alerts lack contextual information, and the formats are not standardized. Without proper preparation such as verification, enrichment, or standardization performed before the analysis, the algorithms that process the alerts need to increase its complexity, i.e. to filter false alerts, while at the same time performing the analysis. As a result, perhaps effectiveness of the analysis methods and accuracy of the final analysis result are affected.

As a quick solution, some systems implement simple preparation, either verification or standardization within their system before the particular analysis method is performed. This practice might improve the accuracy of the final analysis results but as it is not the main focus of the system, the verification or standardization capability might be limited and just suitable to be used within the systems. In other scenario, since there is no alert quality management framework available to guide high-level alert analysis, some of the systems at this level just process the raw alerts directly. Others cannot afford to slot in the preparation process due to commitment to the primary technique implemented. Thus, to help these high-level alert analysis systems reduce complexity, increase effectiveness, and produce more accurate results, a properly planned low-level alert preparation framework that focuses on verifying, enriching, and standardizing IDS alerts is proposed.

In short, this problem statement highlights the current low data quality issue during the production and analysis levels. Observing this issue from the data quality management practice, we believe that the complexity at high-level alert analysis stage can be reduced if the input alerts are well-prepared by properly enriching the alerts with contextual information for verification. We identified a gap between the low-level and the high-level stage, where the verification, enrichment, and standardization process may be performed. Figure 1.1 shows the alert data quality problem overview during the production and the

analysis level. The figure also illustrates the existing process elements and flow while the gap where our problem statement is located is shown by the dashed shaded box.

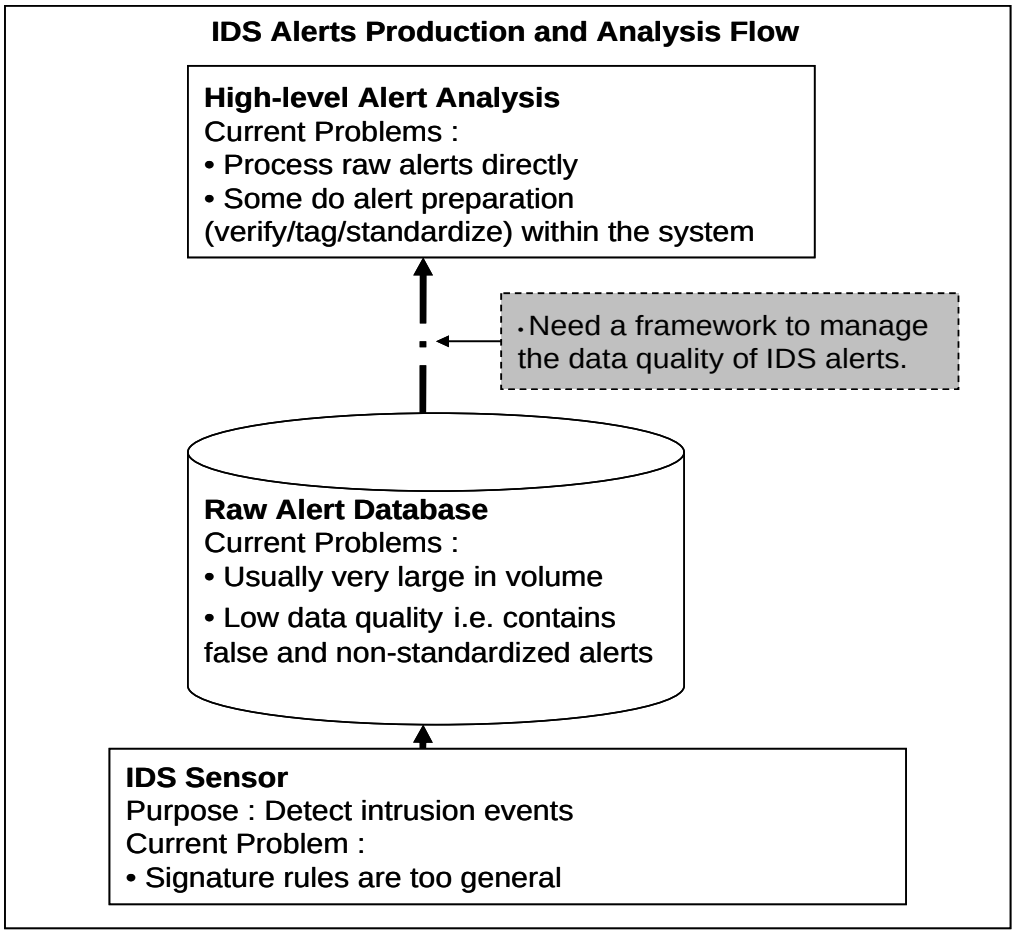

Figure 1.1: Alert data quality problem overview during production and analysis level.

1.3 Proposed Solution: Implementing Intrusion Alert Quality Framework

In this thesis, we propose a solution to address the low data quality IDS alerts problem discussed in Section 1.2 above. The research conducted is to prove the hypothesis that the low data quality alerts can be improved by applying data quality management towards IDS alerts at low-level alert preparation stage. Thus, the two objectives of this research are:

1. To design Intrusion Alert Quality Framework (IAQF) that verifies alert accuracy, enriches alerts with contextual information, and standardizes alert format.
2. To implement prototype of IAQF to show the effectiveness and accuracy of the verified and enriched alerts in helping to reduce false alerts at the analysis stage.

1.3.1 Scope

This thesis focuses on solving the low data quality alert problem. Thus, we concentrate on designing the proper low-level alert preparation of intrusion alert analysis. We leave the high-level alert analysis (correlation, machine learning, or data mining) to perform their specific algorithms and techniques to achieve their individual goal (identify true alerts, analyze, understand, or build attack scenario). Later, these systems have the option to take the output (high data quality alerts) of our low-level alert preparation as their input to indirectly help them produce more accurate analysis. Figure 1.1 has shown the problem overview and research scope where the grey box illustrates our contribution area.

There are some areas that we did not focus on, like the alerts collection mechanism, the contextual (system or network) information gathering tools, the vulnerabilities information storing, and the standardization data format. However, these elements are closely related and directly used in the prototype system to support our framework. The existing techniques are used to collect the alerts from the sensors, to actively and passively scan the network to gather hosts profiles, and to gather vulnerabilities information from vulnerability resources such as Snort signature documentation (Snort, 2007), CERT (CERT, 2007) and CVE Mitre (CVE, 2007). Finally, the existing IDMEF standard is used for normalizing the alerts and enriching them with data quality assessment information.

1.4 Main Contribution

The main contributions of this research are:

1. Verifying alert accuracy using defined data quality parameters.
2. Enriching alerts with measured data quality scores to prepare them for true and false alert classification at the high-level alert analysis stage.
3. Standardizing the enriched alerts using IDMEF standard format to be flexibly used by any high-level alert analysis.

The contributions of this thesis in relation to the existing works in the area of IDS false alerts reduction and data quality are presented in Figure 1.2. The focused area and related works are briefly explained in Chapter 2 to provide background and literature review for this thesis.

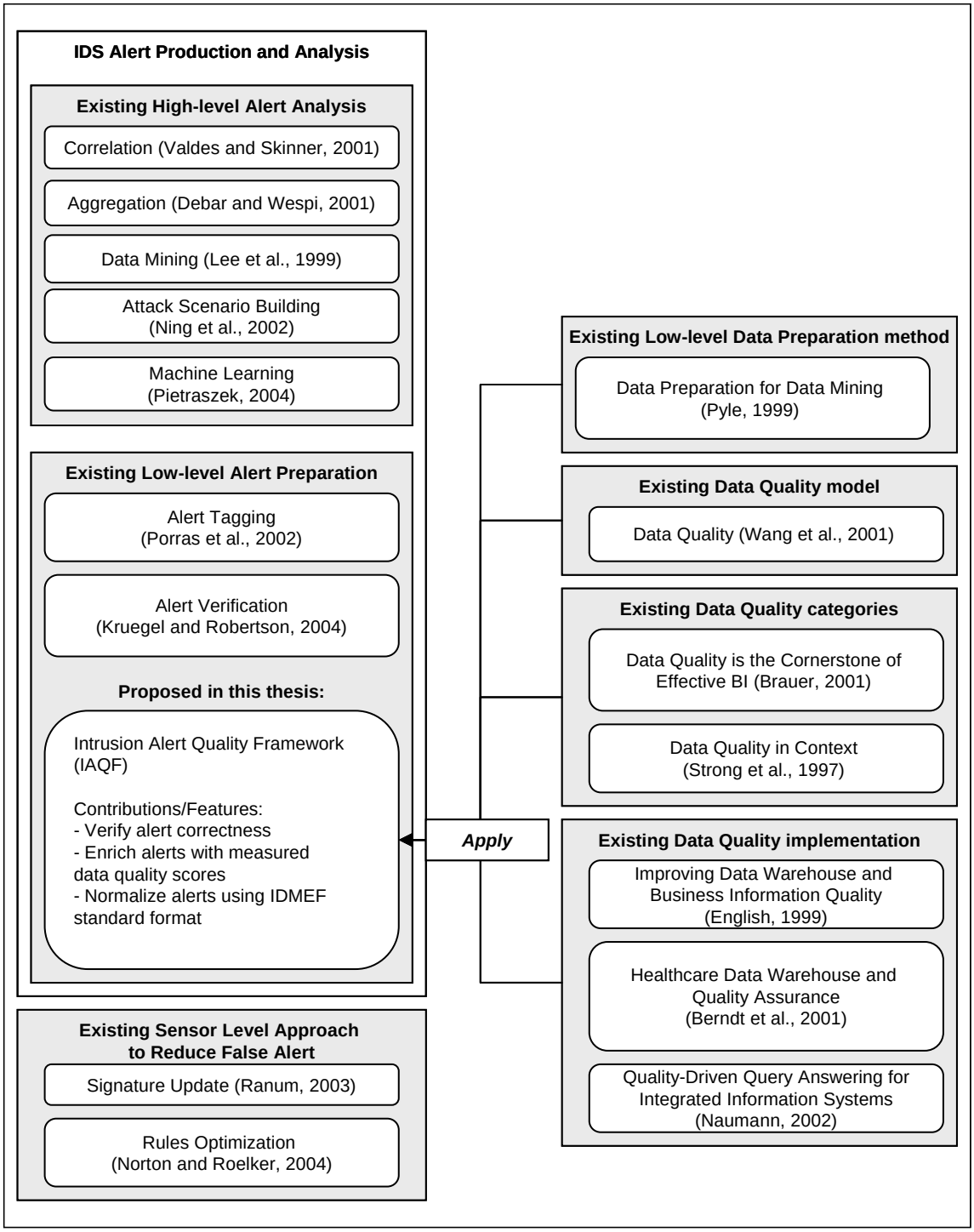

Figure 1.2: The relation of the thesis contributions to the IDS analysis and the data quality domain.

1.5 Terminology

In this thesis, we use terminologies that may have different meaning in other areas of studies. Thus, the explanations about the terminologies used are depicted below:

Security Alert refers to alarm, log or warning report triggered by signature-based IDS sensor when monitored network packet matches sensor's attack signature

False alert refers to inaccurate alerts such as false positive and noises generated by IDS.

True alert refers to accurate alerts that matched all the conditions for the attack events to be successful.

Low-level alert preparation refers to the stage where the alerts are verified and enriched using additional contextual information and standardized to prepare the alerts.

High-level alert analysis refers to the stage where the alerts are processed using sophisticated algorithm such as correlation, aggregation, data mining, or machine learning for false alert reduction, analysis and security decision making.

Low data quality refers to inaccurate or false data. This group of data receives low data quality scores when measured using predefined data quality parameters.

High data quality refers to alerts that are verified to be accurate (true) and receive high data quality scores when measured using predefined data quality parameters. The alerts are enriched with data quality information and standardized.

Data Quality Parameter refers to data quality criteria defined to measure the accuracy of the alerts generated by IDS and meet the high-level alert analysis needs.

Data Quality Score refers to data quality assessment scores measured using predefined data quality parameters, rules, and weights.

Supporting contextual information refers to information that reflect real life conditions of the alert attributes during the attack events, such as vulnerabilities that are going to be affected by the events as well as the hosts' and sensors' profiles in the network where the alerts are triggered.

Precision refers to fraction of the previously known false alerts which has been classified into false alert group.

Recall refers to fraction of the classified false alerts which has been known as false alerts.

1.6 Thesis Outline

This thesis consists of six chapters. In this chapter, motivation, problem statement, objectives, scope, and main contributions of this research are highlighted. In Chapter 2, research background and related works are briefly elaborated. The background thoroughly explains two major areas involved in this research. They are IDS alert analysis and data quality management. The related works are three levels of existing approach to solve false alert problems; sensor level, preparation level, and analysis level. We focus on highlighting two implementations at the alert preparation level: alert verification and alert tagging. These are the works closest to our framework implementation.

Chapter 3 explains in detail the design and architecture of the proposed solution, IAQF. We identify two distinct stages of intrusion alert analysis: low-level alert preparation and high-level alert analysis. The IAQF that applies data quality management is implemented at the low-level data preparation stage and prepare alerts for high-level alert analysis stage. The rest of the chapter covers the detail explanation about the framework components and potential benefits of the approach towards intrusion alert analysis.

Chapter 4 describes the sample prototype implementation of the proposed IAQF. Every component of IAQF, pseudocodes, and sample of alert data quality scores measurements are presented.

Chapter 5 presents the experiments conducted in three case studies, the reduction and scalability results, the results validation, and the evaluation of IAQF effectiveness and accuracy. Three datasets were tested: DARPA 2000, CS USM real LAN, and HoneyNet network traffic.

Finally, Chapter 6 concludes the thesis, reviews the objectives, discusses the potential benefits of IAQF, and finally provides the future works to improve the prototype system.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

In this chapter, background for every domains related to this research are reviewed. This research merges two separate domains, data quality from information management area and intrusion detection from ICT security area. To review ICT security area, where this research belonged, we further discuss the IDS's false alert problem and critically analyze the existing solutions for the problem. We also explain the data quality fundamental concept and model to be used in our approach to solve the IDS problem.

Section 2.2 discusses the IDS and the false alert problem. While Section 2.3 details out the current solutions for the problem, Section 2.4 briefly explains data quality field and model applied in this research to solve the false alert problem. Finally, Section 2.5 summarizes the chapter.

2.2 Intrusion Detection System (IDS)

Intrusion detection is one of ICT security processes executed in a cycle. ICT security is defined as "the process of maintaining an acceptable level of perceived risk" (Bejtlich, 2004). This definition reveals that security is an on going process and not only a one-time action. The ICT security process cycle include four processes; *assessment*, *protection*, *detection*, and *response* process. First, the assessment process covers the policy setting, budgeting, and managing the security implementation. Second, the protection process applies countermeasures to limit the number of attack occurred. Next, detection is the process of analyzing incidents and finally, the response is the recovery steps that are performed after attack occurred (Bejtlich, 2004).

The thesis concentrates on the detection process, or intrusion detection. The intrusion detection is defined as “the process of identifying and responding to malicious activity targeting at computing and network resources” (Amoroso, 1999). Typical enterprise security systems implement defense-in-depth strategy where security devices such as packet filtering routers, stateful firewall, proxy firewall, and IDS are deployed in various strategic locations in the network infrastructure to detect intrusions. Those security devices act as sensors that silently monitor network packets and generate alerts or produce logs when suspicious packets are seen (Northcutt et al.,2003).

There are two types of IDS sensor: network-based (NIDS) and host-based (HIDS) sensor. NIDS monitors all traffic in the monitored network while HIDS monitors the host’s related security information such as application logs, system activities, and file system modification logs. There are also two types of IDS detection mechanisms: signature-based and anomaly-based detections. Signature-based IDS compares pattern in the monitored network packets with a list of attack patterns (signatures) used by the sensors, while anomaly-based IDS monitors system activities and classify them as normal or anomalous. The classification is based on the normal activities monitored by the sensor within specified time duration as the sensor’s learning process prior to the real monitoring usage.

Table 2.1 shows the overview or roadmap of researches that have been done thus far in ICT security area (School of Computer Sciences, USM, 2005). The area covered by our study, log/alert analysis to reduce false alert is highlighted by the dotted-circle. This highlight is to show in which area our research contributes to the ICT security domain in general. As this section explains and shows where our contribution is located in the context of ICT security, the next section further elaborates the problem statement previously stated in Chapter 1.

Table 2.1: Roadmap of researches that have been done in ICT security area (School of Computer Sciences, USM, 2005).

SPECIFIC APPLICATION ORIENTED	**TRUST**	**CONFIDENTIALITY**	**ABUSE**	**ANALYSIS**
Enterprise	-Digital Signature -Public key infrastructure	-Enterprise level security -Agent-Server Security -Radius/Karberos -Honeypot/Honeynet	-Man-in-the-Middle (MIM) -DoS/DDoS -Virus/Worm, Spam -Drone Armies	-Forensics -Enterprise Audit -Enterprise PenTest
Applications	-Biometrics -Smart Card -One time password	-Database security -Web-based Application Security -SSL, SSH	-Buffer Overflow -Format String -Client-side (XST,XSS) -SQL injection -Phising	-Appl. Forensics -Appl. Audit -Appl. Pentest
Data	-Authentication -Non-repudiation -Integrity -Tripwire	-Cryptography (inc. encryption, braid) -steganography -parallelising crypto operations -video/image security	-Packet Spoofing -Cryptanalysis -Brute force -ISN Predictions -Cache Poisoning	-Data Forensics -Log/Alert Analysis -False Positive Reduction
OS Drivers Registeries Interface	-Network security -Mobile Ipv6 security -Tunneling	-IPSec -VPN -Firewall -Intrusion Prevention -Trusted OS	-Rootkit -Trojan horse -OS Fingerprinting -Sniffing -Hijacking -Rerouting	-OS Forensics -OS PenTest -Intrusion Detection
	PROTECTION			

2.2.1 False Alerts Generated by IDS Sensor

This research looks at the false alert problem faced by IDS. To better understand the problem, we classify alerts into *false* and *true* categories. True alert is defined by (Ranum, 2003) as “an alarm that identifies a system that has just been successfully attacked” whereas according to (Timm, 2001), false alerts can be any of the divisions depicted here:

- *Reactionary traffic alerts*; triggered because sensors see a lot of destination unreachable packets as a result of hardware failure
- *Equipment-related alerts*; triggered by load balancer when sensors see unrecognized packets generated by network devices
- *Protocol violations*; triggered because of bugs in software or applications
- *True false positives (benign trigger)*; triggered when sensors make mistake and wrongly interpret a non-malicious event as an attack since it matches the signatures
- *Non malicious alarms (noise)*; triggered when sensors do not make mistake and correctly trigger alert as such intrusion occurred and match the signature but the target hosts are not vulnerable to the attack

Note that the term “alarm” and “alert” are synonymous and used interchangeably in security field.

As mentioned in problem statement in Chapter 1, false alerts exist in any IDS especially signature-based because of signature rules that are too general in an attempt to avoid false negative. Realizing that this problem in any IDS (especially signature-based IDS) is unavoidable, a lot of solutions have been proposed and implemented to eliminate or at least reduce the amount of false alerts so that true alerts can be interpreted accurately by security analysts manually, or with the help of high-level analysis. The existing solutions to handle this false alert problem are explained in the following section.

2.3 Current Solutions for Low Data Quality Alert Problems

The existing solutions to address the low data quality alert problem (especially false alert) in intrusion detection area can be classified into three levels: sensor-level, low-level alert preparation, and high-level alert analysis. We discuss each level in the following subsections and concentrate more on current methods used at low-level alert preparations since here is where our solution is implemented. Figure 2.1 below shows the existing three levels solutions and highlights where our contribution fit in.

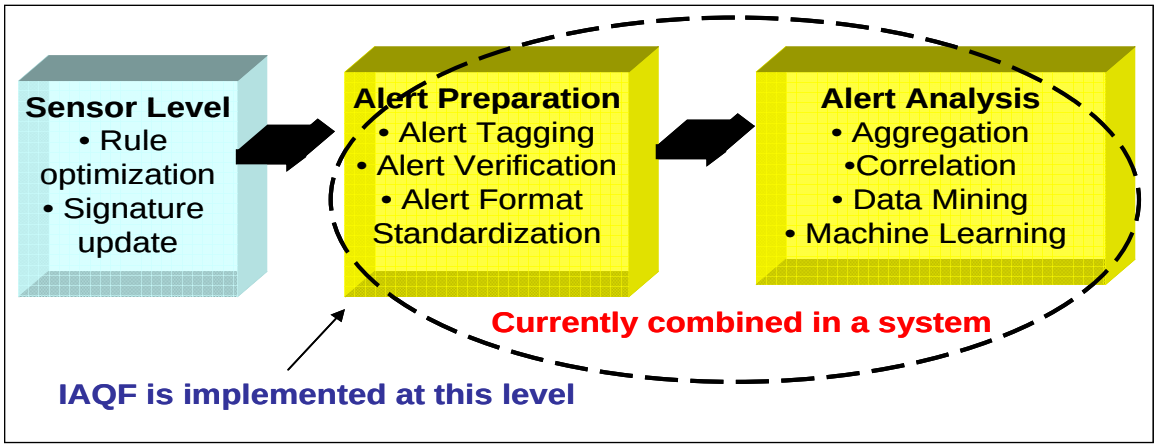

Figure 2.1: Existing solutions for false alert and low data quality alert problems in intrusion detection.

2.3.1 Sensor Level

The first solution to false alert problem is alert reduction at the sensor level. At this level, any or combination of these solutions have been used to improve the sensors' intrusion detection mechanism:

- Tuning sensors' signatures (Ranum, 2003)
- Frequently updating sensor rules (Norton and Roelker, 2004)

The above approaches may help decrease the volume of alerts, avoid data explosion, and indirectly improve sensors' output as the reliability and sensitivity of the sensors are increased. By implementing the above solutions, only specific alerts are generated since

some of the signatures are being turned off. From security analysts point of view, this level of solution is sufficient and helps them a lot in reducing their burden to interpret the alerts since the amount to be analyzed are lessen and irrelevant data are marginally discarded. However, in reality, those solutions are still far from ideal and only act as a short term basic solution in security implementation. Since the network is dynamic and continuously changing from time to time, the tuning also needs to be done every time the network changes. These solutions, most of the time need to be combined with high-level solutions that have broader view of the alert context and not only depends on the reliability and the sensitivity of the sensor. Normally, the alerts still need to be further analyzed, mined, correlated, and filtered to identify the true alerts among the remaining false alerts. The sensor itself can also further improve the alerts by taking into consideration the supporting contextual information available, gathered from the network.

2.3.2 Low-level Alert Preparation

The second level of solution to low data quality alerts problem is at low-level alert preparation stage. At this level, currently, there are several methods implemented to address low data quality alerts: *alert verification*, *alert tagging*, and *alert standardization*. These alert preparation methods are the existing implemented solutions closest to our work. These approaches take the low data quality alerts collected from the sensor and prepare the alerts for high-level analysis. So far, these approaches have been combined in the analysis systems itself (referring to Figure 2.1).

Data preparation such as data enhancement and enrichment at the lower-level has been proven to benefit high-level data analysis method like data mining (Pyle, 1999). Our survey towards several analysis systems also shows that almost all of these systems implement alert preparation methods (either alert verification, alert tagging, or alert

standardization) before they implement their core analysis methods (correlation, aggregation or data mining). Table 2.2 shows the alert preparation methods implemented as well as their benefits for high-level alert analysis. This survey shows that alert preparation is generally needed prior to the high-level analysis techniques. One of the surveyed techniques, the machine learning process, did not implement any alert preparation (refer to the last row of Table 2.2), hence the complexity of the machine learning process is high (Pietraszek, 2004). The level of complexity might be able to be reduced if the alert preparation was implemented prior to the main process.

Alert Tagging

The first low-level data preparation is tagging. This approach was implemented by a particular high-level analysis called M-Correlator (Porras et al., 2002). Alerts are tagged with calculated relevancy score; the attributes are OS type and version, hardware type, service suite, enabled network service, and application. After the alerts are tagged, they are ranked into priority levels at the initial stage of their correlation system. This method has been shown to improve the system's detection rate. This improvement proves that low-level preparation system is indeed needed before high-level analysis methods are being executed.

This alert score measurement is very close to our data quality measurement technique. However, our measurement implementation is part of the IAQF that implement data quality model called TDQM. The advantage of using this model is that the parameters to determine the data quality scores can be redefined, added, or removed. The IAQF is also expandable and flexible for any of the analysis system.

Table 2.2: Data preparation methods implemented in high-level alert analysis systems.

High-level Alert Analysis Systems	Data Preparation Method	Benefits
ACC (Debar and Wespi, 2001)	Associates each alert with a confidence value according to intrinsic and relative inaccuracy factors.	Identifies accuracy.
EMERALD (Porras and Neumann, 1997)	Normalizes alerts using generic resource object.	Provide operating parameters for analysis targets definition, reusability and configuration tuning.
M-Correlator (Porras et al.,2002)	Alert tagging where relevancy and priority of the alerts are scored and tagged.	Measure relevancy and priority of the input alerts.
Fusing a Heterogeneous Alert Stream into Scenarios (Dain and Cunningham, 2001)	Normalizes alert using standard format.	Easily accessed by fusion system.
TIAA (Ning et al., 2002)	Resolves naming inconsistencies.	Easily accessed by alert correlation.
CRIM, MIRADOR (Cuppens, 2001)	Normalizes using IDMEF format.	Easily accessed by clustering function.
A Data Mining Framework for Building Intrusion Detection Models (Lee et al., 1999)	Computes accurate models from very large amount of input data using learning agents and classify the input data using detection agent.	Lightweight detection where the heavy tasks are processed by learning agents.
Comprehensive Approach to Intrusion Detection Alert Correlation (Valeur et al. 2004)	Alert verification and tagging by determining the success of the intrusion attempt.	The verification prepares the data for the correlation with success information.
Machine Learning (Pietraszek, 2004)	Not implemented.	Increase complexity of the machine learning

Alert Verification

The second method implemented at low-level alert preparation stage is verification. This approach verifies IDS alerts by integrating them with contextual or supporting information collected from network or system environment where the alerts are triggered. There are three types of network or system information gathering techniques: *passive scanning*, *active scanning*, and *post-attack verification* techniques.

First, using *passive scanning*, the system and network information such as running OS and running services are gathered by monitoring the network silently without sending any packets to the network. An example of a product that implements this passive scanning technique is Passive Vulnerability Scanner (PVS) (Tenable Network Security, 2007).

The second type of contextual information gathering technique is the *active scanning*. Usually, this technique is done at regular interval from time to time. These gathered information are stored in a database form time to time and are available during the alert analysis. Nmap (Insecure.org, 2007) is an example of a security scanner that uses the active scanning method.

The third type of information gathering and context integration is the *post-attack verification*. This technique investigates target hosts to discover forensic traces and evidence after alerts are generated. These evidences are used to support the hypothesis of whether the alerts were successful or not. This verification system was implemented by (Kruegel and Robertson, 2004) and aimed at determining alert success status by considering real-time network information. The alerts are marked with “successful”, “unsuccessful”, and “undetermined” tags.

Active and passive scanning, each has its constraints and accuracy issues. Thus, the best solution for the time being is to combine both techniques as implemented by RNA (Real-time Network Awareness) (Shenk and Shackleford, 2007), a commercial tools from Sourcefire. RNA is built to fulfill the need of system and network intelligence to make security analysts aware of what systems they are currently protecting. This real-time knowledge about the systems plugged in and out of the network is highly important to

analyze IDS alerts since it will determine whether the alerts are true or false. Some of the alert supporting information that can be gathered using RNA are OS and host's alive status using passive discovery, opened ports using active discovery, anomaly detection, and host criticality information.

In this research, we also test our proposed framework in real network using combination of the active and passive scanning. The active scanning was done using Nmap while the passive scanning was performed using p0f (Zalewski, 2006) and PADS (Shelton, 2005). P0f was used to do real time OS fingerprinting while PADS was used to detect hosts that exists in the network as well as the running services.

Alert Format Standardization

Besides tagging and verification, another preparation method is format standardization. The format agreed by the security community is used to standardize the alerts coming from heterogeneous security data sources. The alerts are standardized prior to being processed by the high-level analysis such as correlation and aggregation. It is an attempt to solve part of low data quality issue which is the non-standardized alerts or logs produced by different IDS vendors. There have been several formats proposed by the intrusion detection community such as IDMEF (Debar et al., 2006) and CIDF (Staniford-Chen et al., 1998) where each format is designed for different purposes. Without a common format agreed upon, effective efforts to aggregate and correlate logs or alerts can be a daunting task.

IDMEF was introduced by Debar et al. (2006) as a standard data format to present alerts generated by IDS. Figure 2.2 shows the simplified version of the IDMEF model (work in progress) as of September 17, 2006. The purpose of this data model is to provide a

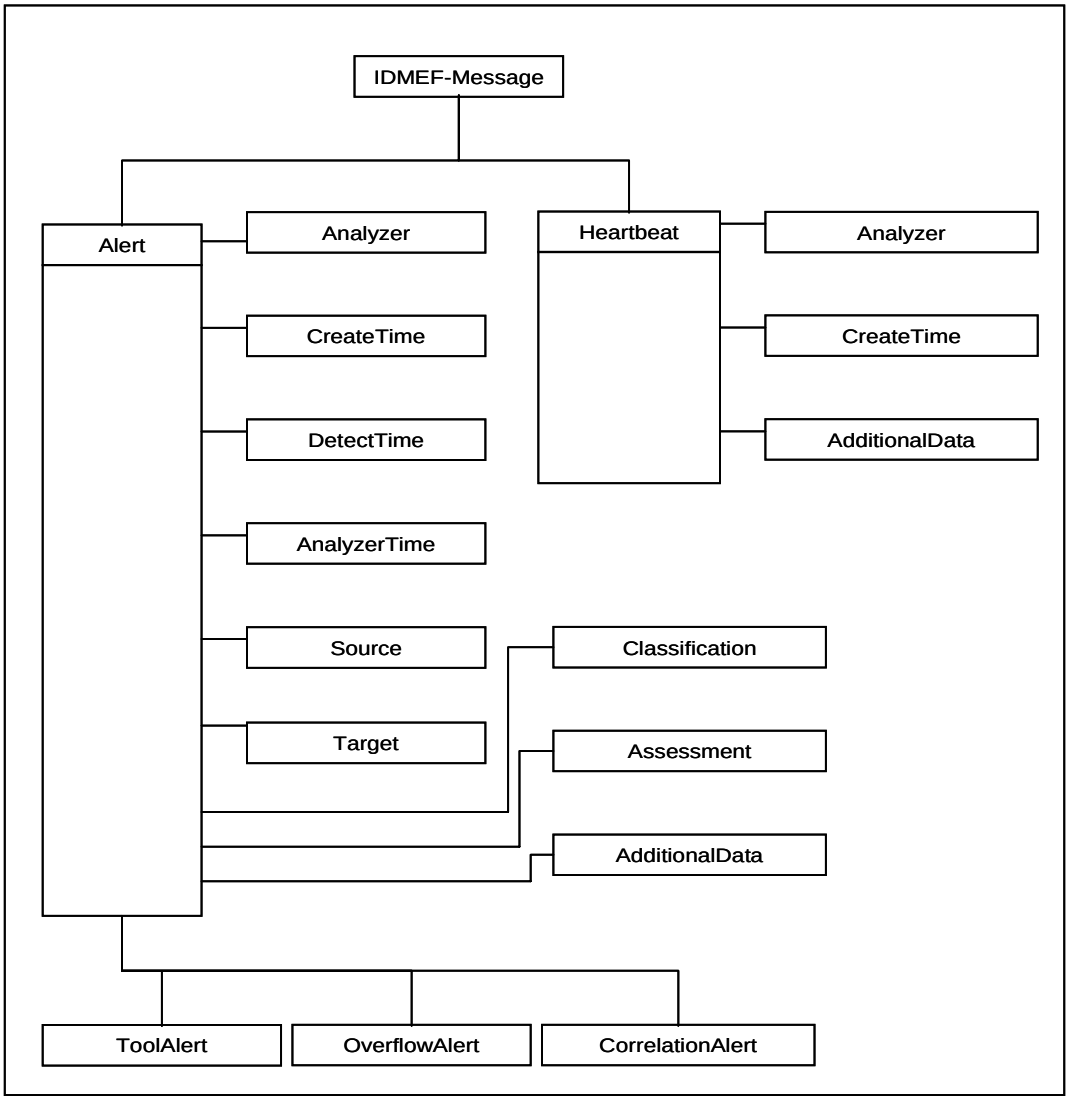

Figure 2.2: A simplified version of the IDMEF model (work in progress) as of September 17, 2006

.

standard representation of the alerts (simple or complex) reported by IDS. An example of an analysis system that implements IDMEF within the system to normalize it's raw alerts input collected from IDS sensors is CRIM of MIRADOR project (Cuppens, 2001) (refer Table 2.2).

IDMEF has several strong points that can be exploited to solve alerts standardization and matching issues. IDMEF data model addresses several IDS alert representation problems that are:

- Heterogeneous alert information; the object-oriented implementation is extensible and flexible enough to cater alerts with simple or detail information.
- Different IDS environments; there are support classes that handle varieties of data sources such as detection based on network traffic, OS logs or audit data etc.
- Different sensors capabilities (lightweight or complex); extensions can be done using subclassing or association of new classes.
- Different operating environments (network or OS used); subclasses can be used to add additional attributes to accommodate different characteristics of reported attacks.
- Different commercial vendor's objectives; the object-oriented inherent features takes care of this problem. (Debar et al., 2006)

Another data model called M2D2 was proposed by Morin et al. (2002). M2D2 is formally defined to model relevant data for alert correlation. The model includes four types of information: information system characteristics, vulnerabilities, security tools, and alert events. The model is quite comprehensive and able to enrich IDMEF formatted alerts with this information. However, the model only intends to provide the four types of information and is not extensible; the data quality assessment of the alerts has not been planned to be included. Thus, our framework may be implemented on top of the model to further verify and enrich the alerts with data quality information (in the form of data quality assessment scores).

2.3.3 High-level Alert Analysis

After explaining the two levels of false alert solution (sensor level and low-level alert preparation), we finally describe the third level of false alert solution that is the high-level alert analysis (refer Figure 2.1). Since IDS normally produces thousands of alerts per day, it is difficult to analyze the alerts manually. Therefore, at this level, methods such as correlation, aggregation, data mining or machine learning are used to better understand attack scenario and at the same time filter or reduce false alerts.

There have been a lot of high-level analysis systems developed (refer Table 2.2). The alerts aggregation and correlation techniques were implemented by IBM (Debar and Wespi, 2001) to find duplicates and group alerts according to predefined criteria. Lee et al. (1999) uses data mining to classify and correlate alerts while SRI International (Porras et al., 2002) used probabilistic method to develop EMERALD that parse, filter, format, analyze, and correlate alerts. EMERALD was then further enhanced and called M-Correlator (refer Section 2.3.2). Another high-level alert analysis developed is an attack scenario builder called TIAA (Toolkit for Intrusion Alert Analysis) (Ning et al., 2004) which processes alerts and produces attack scenario based on prerequisite-consequence-based correlation method.

2.3.3.1 Generic High-level Alert Analysis Procedures

Gorton (2003) has done a survey towards the above analysis systems and proposed a generic procedure containing six tasks commonly accomplished by the systems that uses Intrusion Alert Correlation (IAC) analysis method. Figure 2.3 below shows the procedures. The detail about every task is elaborated in the following paragraphs.