A FALSE ALERT REDUCTION AND AN ALERT SCORE ASSESSMENT FRAMEWORK FOR INTRUSION ALERTS

KARIM HASHIM KRAIDI AL-SAEDI

UNIVERSITI SAINS MALAYSIA
2013

A FALSE ALERT REDUCTION AND AN ALERT SCORE ASSESSMENT FRAMEWORK FOR INTRUSION ALERTS

By

KARIM HASHIM KRAIDI AL-SAEDI

Thesis submitted in fulfillment of the requirements
for the degree of
Doctor of Philosophy

September 2013

ACKNOWLEDGEMENTS

بسم الله الرحمن الرحيم
{نرفع درجات من نشاء وفوق كل ذي علم عليم "76"} (سورة يوسف)

I would like to take this opportunity to convey my sincere thanks and deepest gratitude to my main supervisor, Prof. Dr. Sureswaran Ramadass, for all the kind assistance and valuable guidance he has kindly provided to me during my studies. I consider myself privileged to have had the opportunity to work under his effective and valuable guidance. My thanks would never end up to this great Prof. as whatever praises being said to him are not enough to be expressed. His morale support is very much valuable to me.

Moreover, I would like to give my deepest thanks to my co-supervisor, Dr. Wafaa A. H. Ali and Mr. Selvakumar Manickamn for all their valuable guidance, comments and support provided to me during my studies. I also would like to express my gratitude to my friend Ass. Prof. Dr. Hussein Al-Khfaji for his non-stop encouragements and guidance provided to me. I would like to convey my appreciation to all NAv6 Centre members and the Institute of Postgraduate Studies. I would also like to thank USM for the attention and assistance they have given me from my start of pursuing this academic programme.

I would like to express my gratitude to my dear parents; my kindness father and mother, my brother Kamil, my wonderful wife, and my sons Haider, Noor Aldean, Bahaa Aldean and Ali Aldean for giving me their great love, endless support, sympathy and encouragement during every step of my study. Last but not least, a big thanks goes to Al-Mustansiriyah University for giving me the opportunity to leave for the favour of completing my PhD degree, especially, the lecturers and staff in Computer Science Department and College of Science. I would like to thank all my friends in Malaysia and Iraq for their kind support and marvelous assistance.

The favor, above all, before all, and after all, is entirely Allah's, to whom my never-ending thanks and praise are humbly due.

Thank you!

Karim Hashim Kraidi Al-Saedi
Penang, Malaysia, March 2013

TABLE OF CONTENTS

Pages

ACKNOWLEDGMENTS.. ii

TABLE OF CONTENTS ……………………………………………. iii

LIST OF TABLES …………………………………………………. ix

LIST OF FIGURES ………………………………………………….. xii

LIST OF ABBREVIATIONS …………………………………...... xv

ABSTRAK …………………………………………………………. xvii

ABSTRACT ………………………………………………………….. xix

CHAPTER 1: INTRODUCTION

1.1 Introduction ………………………………………………... 1

1.2 Background ……………………………………………….... 2

1.3 Alert Detection Engines (ADE) …………………………... 2

1.4 Problem Statement ……………………………………….... 5

1.5 Research Objectives ………………………………………… 6

1.6 Thesis Contributions ………………………………………… 7

1.7 Research Methodology ……………………………………… 9

1.8 Thesis Outline ……………………………………………….. 10

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction ………………………………………………... 11

2.2 Background ………………………………………………... 11

2.2.1 Detection Engines …………………………………… 11

2.2.2 Alert Detection Engine Classification …………………….. 12
2.2.2.1 Information Source ………………………………... 13
2.2.2.2 Type Analysis …………………………………... 15
2.2.2.3 Response ………………………………………… 16
2.2.2.4 Detection Time ……………………………….. 17
2.2.3 Snort ……………………………………………….. 18
2.2.4 Alert Log Feature Set ……………………………… 19
2.2.4.1 Common Alert Feature Sets ………………………. 19
2.2.4.2 Time Feature ……………………………………… 21
2.2.4.3 Common Vulnerabilities And Exposures Feature 22
2.2.5 Association Rules……………………………………….. 23
2.2.5.1 Association Rules Requirements……….............. 25
2.2.5.2 Apriori Algorithm ...……………………………… 28
2.2.5.3 Association Rules Generation ...…………………. 32
2.3 Information Gain Ratio Algorithm ……………………………… 37
2.4 Related Work …………………………………………………. 38
2.4.1 Alert Reduction ...……...………………………………... 38
2.4.1.1 Alert Classification Techniques...…………………. 39
2.4.1.2 Alert Correlation Techniques… 44
2.4.2 Alerts Assessment ...…………………………..................... 48
2.5 Comparison of Related Work with the Proposed Framework 52
2.6 Chapter Summary………………………….......................... 56

CHAPTER 3: PROPOSED FRAMEWORK

3.1 Introduction …………………………………………….......... 57

3.2 Overall Architecture of the Proposed Framework........................ 59
3.3 Traffic Data Retrieval And Collection Mechanism Module ……… 64
3.3.1 Pre-Knowledge……………………………………………..... 66
3.3.1.1 Procurement of Alerts ……………………………… 66
3.3.1.2 Field Reduction And Data Standardization of Alerts 67
3.3.2 Feature Extraction of Alerts …………………………….... 70
3.3.3 CVE DB……………………………………………………. 70
3.4 False Alert Reduction Module…………………………………… 70
3.4.1 Counter of Calculating Alerts Component……………….. 73
3.4.2 Reduction of Alert Component ……………………….…..... 73
3.4.2.1 Similarity Alert Process …………………………..... 74
3.4.2.2 Similarity Alert Based on Time Threshold Process 74
3.4.2.3 Related False Positive Alerts Process …………….. 74
3.4.3 A Counter to Calculate Grouping of Alerts……….......... 77
3.5 Alert Score Assessment (ASA) Module ………………..……. 77
3.5.1 Generating Featuresets…………………………………….. 79
3.5.1.1 1-Featuresets…………………………………… 80
3.5.1.2 2-Featuresets…………………………………… 81
3.5.1.3 K-Featuresets…………………………………… 82
3.5.2 Generating Rules for Featuresets and Auto Threat Scores... 82
3.5.2.1 Generate Rules For Featuresets ………..………….. 83
3.5.2.2 Threat Score of Features …………………………... 83
3.5.2.3 Threat Score of Alert …………………………….. 84
3.6 Chapter Summary …………………………………………… 85

CHAPTER 4: PROPOSED FRAMEWORK IMPLEMENTATION

4.1 Introduction………………………………………………………. 86

4.2 Overall Implementation Details of the Proposed Framework..... 86

4.2.1 Challenges in the Programming Language ………………. 87

4.2.2 Scalability of the Proposed Framework ………………….. 87

4.3 Implementation of the Proposed Framework …………………... 88

4.4 Implementation of Traffic Data Retrieval and Collection Mechanism Module 90

4.5 Implementation of False Alert Reduction Module………………. 94

4.6 Implementation of Alert Score Assessment (ASA) Module ……. 98

4.6.1 Multi Feature Apriori (MFA) Algorithm …………………. 98

4.6.1.1 Formal Concepts of the Featureset Problem ……... 100

4.6.1.2 1-Featuresets Generation…………………………... 103

4.6.1.3 2-Featuresets Generation…………………………... 104

4.6.1.4 K-Featuresets Generation………………………….. 106

4.6.1.5 Complexity………………………………………… 108

4.6.2 Improved Rules Generation (IRG) algorithm …………… 109

4.6.2.1 First Phase: Generate Rules Featuresets………….. 110

4.6.2.2 Second Phase: Extraction of the Threat Score of Features 112

4.6.2.3 Third Phase: Auto Threat Score of the Alert ……. 113

4.6.3 Comparisons between the Assessment Algorithms and Original Algorithms 114

4.6.3.1 Apriori Algorithm and MFA Algorithm ……… 114

4.6.3.2 Rules Generation Algorithm and IRG Algorithm 115

4.8 Chapter Summary ……………………………………………… 117

CHAPTER 5: RESULT AND DISCUSSION

5.1 Introduction ………………………………………………………… 119

5.2 Experiment Environment ………………………………………... 119

5.2.1 Proposed Framework Hardware Specifications……………. 120

5.2.2 Proposed Framework Software Specifications…………...... 120

5.3 Datasets …..………………………………………………………… 120

5.3.1 DARPA 1999 Dataset ……………………………………... 121

5.3.2 NAv6 2012 Dataset ………………………………………... 121

5.4 Evaluation Criteria of the Proposed Framework ……………….. 123

5.4.1 Multiple Features ………………………………………...... 123

5.4.2 Reduction Rate ……………………………………….…..... 125

5.4.3 Accuracy ………………………………………………….. 126

5.5 Evaluations of the Traffic Data Retrieval and Collection Mechanism Module 127

5.5.1 Experiments of the Traffic Data Retrieval and Collection Mechanism Module 127

5.6 Evaluations of False Alert Reduction Module ………………… 131

5.6.1 Experiments of False Alert Reduction Module ………… 131

5.6.2 Comparisons with Other Reduction Approaches ……….. 137

5.7 Evaluations of the Alert Score Assessment Module ………….. 143

5.7.1 Experiments of the Alert Score Assessment Module …… 143

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1 Introduction ………………………………………………………… 151

6.2 Conclusions ………………………………………………………… 151

6.3 Future Work ………………………………………………………….. 153

REFERENCES ………………………………………………………… 154

LIST of PUBLICATIONS ……………………………………………. 155

LIST OF TABLES

Table 2.1 The Items Abbreviations of Database 35

Table 2.2 Transaction Database 35

Table 2.3 Large Itemsets with Minsup = 2 36

Table 2.4 Association Rules 36

Table 2.5 Information Gain Ratio on Alerts Features 38

Table 2.6 Summary of the Related Works Compared with the Proposed Approach for Reduction 55

Table 2.7 Summary of the Related Works Compared with the Proposed Approach for Assessment 56

Table 3.1 The Rules Used in the NAR Algorithm 75

Table 3.2 Section of the 1-Featureset 80

Table 3.3 Section of the 2-Featureset for (IPs) Feature 81

Table 3.4 Section of the 3-Featureset for (IPs) Feature 82

Table 4.1 Result of Step 2 93

Table 4.2 Result of Step 3 93

Table 4.3 Result of Step 4 94

Table 4.4 The Rules Used in the NAR Algorithm with Threshold of CVE 95

Table 4.5 The Example of a Few Alerts 96

Table 4.6 The Result of First Phase 96

Table 4.7 The Result of Second Phase 97

Table 4.8 The Result of Third Phase 98

Table 4.9 The 1-Featureset 105

Table 4.10 The 2-Featureset 105

Table 4.11 The 3-Featureset 107

Table 4.12 The 4-Featureset 107

Table 4.13 The Featureset Using the Apriori Algorithm 114

Table 4.14 Generating Rules by Original Rules Generation Algorithm 115

Table 4.15 Comparisons Between the Apriori and MFA Algorithms 116

Table 4.16 Comparisons Between the Rules Generation and IGR Algorithms 116

Table 5.1 Basic Dataset Statistics to be Used 120

Table 5.2 The Sample Amount of Alerts Produced from the False Alert Reduction Module for the DARPA 1999 Dataset 122

Table 5.3 The Comparison of the Flexibility Using Multiple Features 123

Table 5.4 The Entire Feature's Data Types 125

Table 5.5 The Result of the Weighted Features for the DARPA 1999 Dataset 128

Table 5.6 The Result of the Weighted Features for the NAv6 2012 Dataset 130

Table 5.7 Parameters Used in Experiment 1 131

Table 5.8 The Results of the 3 Days of the DARPA 1999 Dataset 132

Table 5.9 The False and Real Alerts of the 3 Days of the DARPA 1999 Dataset 134

Table 5.10 Comparisons Reduction Ratio Between the Proposed Module with Other Approaches by Using Three Days from DARPA 1999 Dataset 134

Table 5.11 Parameters Used in Experiment 2 135

Table 5.12 The Results of the DARPA 1999 for Five Weeks by Using the Proposed Module 136

Table 5.13 The Results of the false and real alerts of the DARPA 1999 for Five Weeks 136

Table 5.14 The Classification of the Reduced Rate Based on the Attack Label 138

Table 5.15 Comparisons Reduction Ratio between the Proposed Module with Another Approach by Using Second Week from DARPA 140

1999 Dataset

Table 5.16 Comparisons Between the Proposed Module and Other Approaches 141

Table 5.17 Parameters Used in Experiment 3 142

Table 5.18 Results of the NAv6 2012 Dataset for Two Weeks by Using the Proposed Module 142

Table 5.19 Parameters Used in Experiment 1 144

Table 5.20 Results of Experiment 1 144

Table 5.21 The Types of Alert with Min and Max Threat Score of the Experiment 1 145

Table 5.22 An Example of the Threat Score of the Alert 146

Table 5.23 Parameters Used in Experiment 2 147

Table 5.24 The Results of Experiment 2 147

Table 5.25 The Types of Alert with Min and Max Threat Score of the Experiment 2 148

Table 5.26 The Comparison Threat Score Results Between Experiment 1 and Experiment 2 149

LIST OF FIGURES

Figure 1.1	Alert detection engine	3
Figure 1.2	Research methodology	9
Figure 2.1	Classifications of the Alert Detection Engine	13
Figure 2.2	Alert in fast mode	18
Figure 2.3	Alert in full mode	19
Figure 2.4	An example of CVE information details	26
Figure 2.5	Apriori algorithm	29
Figure 2.6	Candidate generation function	30
Figure 2.7	Pruning step algorithm	32
Figure 2.8	Rule generation algorithm	33
Figure 2.9	The relation of the IDS and the proposed filtration	40
Figure 2.10	The framework of decision support system	42
Figure 2.11	The framework of the system proposed by Al-Mamory et al.(2010)	45
Figure 2.12	Alert cluster framework	46
Figure 2.13	An overview of assessing the threat of network intrusions	48
Figure 2.14	Overview of the system architecture proposed	50
Figure 2.15	The network threat assessment framework	51
Figure 3.1	The research methodology of the proposed framework	58
Figure 3.2	False alert reduction system architecture	60
Figure 3.3	Alert score assessment system architecture	61
Figure 3.4	False alert reduction and alert score assessment system architecture	62

Figure 3.5 The architecture of traffic data retrieval and collection mechanism 65

Figure 3.6 The contents of the traffic data retrieval and collection mechanism module 66

Figure 3.7 Section from a text file of a snort alert 67

Figure 3.8 Architecture of field reduction and data standardization of alerts 67

Figure 3.9 An example for alerts 69

Figure 3.10 The architecture of the false alert reduction module 72

Figure 3.11 The content of the false alert reduction module 72

Figure 3.12 The reduction of alert component 73

Figure 3.13 The alert score assessment module 78

Figure 3.14 The content of the alert score assessment module 79

Figure 4.1 Flow chart of implementation of traffic data retrieval and collection mechanism module 91

Figure 4.2 New alert reduction algorithm 94

Figure 4.3 The proposed multi feature apriori algorithm 99

Figure 4.4 The MFA mining process 102

Figure 4.5 The result after applying steps 1 and 2 103

Figure 4.6 The improved rules generation algorithm 109

Figure 5.1 The results of the experiment 1 using three days of the DARPA 1999 dataset 132

Figure 5.2 The reduction rate for three days of the DARPA 1999 dataset 133

Figure 5.3 Alert reduction for five weeks of using the DARPA 1999 dataset 139

Figure 5.4 Reduction rate for five weeks of using the DARPA 1999 dataset 140

Figure 5.5 Comparisons of the reduction process between the proposed module and other approaches 141

Figure 5.6 Alert reduction for two weeks using the NAv6 2012 dataset 143

Figure 5.7 Comparison between max alert scores between 2 Experiments 149

LIST OF ABBREVIATIONS

ACARM	Alert Correlation, Assessment and Reaction Module
AD	Affinity Degree
ADE	Alert Detection Engines
ASA	Alert Score Assessment
Conf	Confidence
CVE	Common Vulnerabilities and Exposures
DARPA	Defense Advanced Research Projects Agency
Dconf	Degree of Confidence
DDos	Distribution Denial of Service
DM	Data Mining
Dsupp	Degree of Support
FE	Feature Extraction
GUI	Graphic User Interface
HAF	High Alert Frequency
HADE	Host-Based Alert Detection Engine
HMMs	Hidden Markov Models
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Protection System
IRG	Improve Rules Generation
LAN	Local Area Network
MFA	Multi-Feature Apriori
minconf	minimum confidence
minsup	minimum support
MS-Dos	Microsoft System-Disk Operating System
NADE	Network-Based Alert Detection Engine
NAR	New Alert Reduction
NAv6	National Advanced IPv6 Centre
OS	Operating System
PFAs	Probabilities of having Following Attacks

RFC	Rule File Comes
Rr	Reduction rate
TCP	Transmission Control Protocol
TIDs	Transaction Identifiers
TSA	Threat Score Alert
TSF	Threat Score Feature
UFP	Usual False Positives

KERANGKA PENGURANGAN AMARAN PALSU DAN PENILAIAN SKOR AMARAN UNTUK AMARAN PENCEROBOHAN

ABSTRAK

Pengesanan Enjin Amaran (Alert Detection Engine, ADE) merupakan sistem keselamatan rangkaian yang amat berkuasa. Ia digunakan untuk menjaga keselamatan rangkaian komputer. ADE mengesan pencerobohan keselamatan yang menyebabkan langkah keselamatan tidak dapat dilindungi. Walau bagaimanapun, ADE masih menghadapi masalah penjanaan amaran dengan jumlah amaran positif-palsu yang banyak. Sering kali, ADE menjana amaran yang banyak dimana sebahagiannya adalah benar dan sebahagian lagi adalah palsu berlebihan. Justeru, hal ini menyebabkan bertambahnya kekeliruan bagi kalangan pembuat keputusan, untuk membuat penilaian amaran berkenaan. Secara khususnya, amaran benar daripada ADE tidak dikelaskan berasaskan magnitud yang dihadapi oleh mereka. Hal ini menyukarkan juruanalisis keselamatan untuk mengenal pasti serangan serta mengambil tindakan pemulihan terhadap ancaman. Oleh itu, magnitud bagi setiap ancaman perlu dikategorikan. Pengkategorian darjah ancaman boleh dilakukan dengan menggunakan teknik perlombongan data, terutamanya apabila melibatkan data yang banyak. Terdapat pelbagai pendekatan pengurangan dan penilaian yang dicadangkan bagi menyelasaikan permasalahan ini.

Tesis ini mengemukakan satu kerangka baru, Kerangka Pengurangan Amaran Palsu Dan Penilaian Skor Amaran Untuk Amaran Pencerobohan. Alasan bagi penggunaan kerangka adalah untuk mengurangkan amaran positif-palsu dan menilainya dalam usaha menentukan skor daripada ancaman amaran. Di samping itu, ia juga dapat memberikan pemahaman yang lengkap tentang serangan terhadap rangkaian serta memudahkan proses menganalisis mahupun menjimatkan masa mereka. kerangka ini merupakan suatu sistem sendirian yang boleh berfungsi sebagai dalam-talian altaupun luar-talian. kerangka menggabungjalinkan tiga algoritma. Pertama, algoritma pengurangan baru (NAR) untuk mengeluarkan lebihan amaran daripada fail amaran serta mengurangkan positif-palsu. Kedua, algoritma Apriori multi sifat (MFA) dipertingkat, yang mampu meningkatkan kerja dengan multisifat serta menjana set sifat berdasarkan korelasi pelbagai sifat amaran. Penjanaan set sifat adalah sangat penting dan merupakan asas penilaian. Ketiga, algoritma IRG (improved rules generation), yang diubahsuai bagi membolehkan ia menjana aturan set sifat, mengira skor sifat ancaman, dan menilai seluruh ancaman dengan mengira skor ancaman bagi setiap amaran.

Keputusan kerangka merangkumi dua fasa. Fasa pertama adalah modul pengurangan, yang mengurangkan jumlah amaran positif-palsu sebanyak 92.27% melalui penggunaan set data DARPA 1999 dan 93.39% melalui penggunaan set data NAv6 2012. Fasa kedua adalah modul penilaian iaitu yang mengakses amaran yang masih ada dalam modul pengurangan dan julat skor ancaman amaran adalah 2 – 8.89 daripada 10.

A FALSE ALERT REDUCTION AND AN ALERT SCORE ASSESSMET FRAMEWORK FOR INTRUSION ALERTS

ABSTRACT

The Alert Detection Engine (ADE) is a powerful network security system that is used to secure computer networks. ADE can detect security breaches which other forms of security measures unable to uncover. Yet, it still suffers from the problem of generating huge amounts of alerts that are mostly false positives. Each ADE generates a large number of alerts, where some are real and the others are not (i.e. false or redundant alert). Consequently, this increases the ambiguity among the decision makers as they conduct assessments of alerts. In particular, real alerts of ADE are not classified based on the magnitude of the threat they pose. Therefore, it is difficult for the security analyst to identify attacks and take remedial action against their threats, making it necessary to categorize the magnitude of each threat. For this reason, it becomes necessary to categorize the degrees of threat using data mining techniques, especially where huge data are involved. Several reduction and assessment approaches have been proposed to solve these problems; however, they unable to address many other problems related to ADE.

This thesis proposes a new framework called A False Alert Reduction and an Alert Score Assessment Framework for Intrusion Alerts. The objectives of using this framework are to reduce the false alerts and to assess such alerts and examine their threat scores. This work aims to provide a full understanding of the network attacks as well as ease the process for the analysts and save their time. Framework is a standalone system that can work online and offline. It combines the following algorithms: the first algorithm is New Alert Reduction (NAR) algorithm to remove the redundancy from the alert's file and reduce the false positives. The second algorithm is called the improved Multi Feature Apriori (MFA) algorithm that has been enhanced to work with multi features and generate featuresets based on the correlation of various alert features (these generated featuresets are very important because they are basic requirements in any assessment), and the third algorithm is called the Improved Rules Generation (IRG) algorithm, which has been modified to enable it to generate rules for featuresets, compute the threat scores of the features, and assess the whole threat by calculating the threat score for each alert

The results of framework have two phases. The first phase involves the reduction modules that reduce the amount of false positive alerts by 92.27% and 93.39% using the DARPA 1999 and the NAv6 2012 data sets, respectively. The second phase involves the assessment modules that assess the remaining alerts from the reducing module, the rank threat score of alerts was ranged from 2 to 8.89 scores out of 10 scores.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Numerous techniques and approaches are used to address the threats faced by computer networks todays. The threats discussed within the context of this thesis are those caused by hackers and others trying to attack the network via denial of service methods and/or malware. The techniques used to address there threats can basically be divided into two categories; A ***proactive approach strategy*** to prevent any potential intrusion from befalling in the first place, and a ***reactive approach strategy*** to detect any break-in or intrusion that is in progress.

The “Proactive Approach Strategy” involves setting up firewalls with strong filtering policies and Intrusion Prevention Systems (IPS) to prevent potential attacks. Patching security vulnerabilities in applications is also important within the “Proactive Approach Strategy” (Elshoush & Osman, 2011).

The “Reactive Approach Startegy” involves hackers and intruders who will always try to circumvent the security perimeter set up. Through various methods such as zero-day attacks and social engineering, hackers can possibly infiltrate a network and its computers. Some of these reactive approaches involve Alert Detection Engines (ADE), (including Intrusion Detection System IDS, Firewalls, etc), malware data mining and real-time network monitoring.

1.2 Background

Penetrations and threats are usually made by hackers to get into the desired networks or computer systems and to attack, hit or control the victim either by sending a virus, DDoS, worms, bot, and other forms of malware. (Elshoush & Osman, 2011; Tjhai, 2011).

Most Alert Detection Engines, like an Intrusion Detection System (IDS) are an integral component of a network's security architecture. They monitor incoming packets to try to identify their intrusive behaviors. An alarm is raised if a possible intrusive event is detected. This alarm gives the security analyst the opportunity to react promptly against the possible encountered threat. Most of the outputs of such engines contain a high proportion of unsuccessful alarms known as false positive alerts.

Alert flooding occurs when signature-based Alert Detection Engines produce an alert for all potential malicious packets. While such a statement sounds logically reasonable, the amount of such potential alerts can be overwhelming. Accurate and efficient assessments are required to identify and reduce these false positive alerts. The following section introduces some background information about Alert Detection Engines.

1.3 Alert Detection Engine (ADE)

An Alert Detection Engine can assist in understanding external threats facing the network. Even though the Alert Detection Engine is unable to prevent threats, yet, it collects information when threats are encountered. This collected information could then be used to correct mistakes and fill the gaps within the security architecture of the organisation (Maggi et al., 2009).

Alert Detection Engines (like Intrusion Detection Systems (IDS)) are available in the form of either software or hardware, where in both cases, they are used to monitor the network traffic. In network traffic, information within the network is not transmitted in-line into the alert detection engine device; rather, the alert detection engine monitors the traffic through an out-of-band network interface. When the Alert Detection Engine detects suspicious activity within the network, it sends an alert to the network administration regarding the potential threat, which could be an intrusion attempt (Elshoush & Osman, 2011).

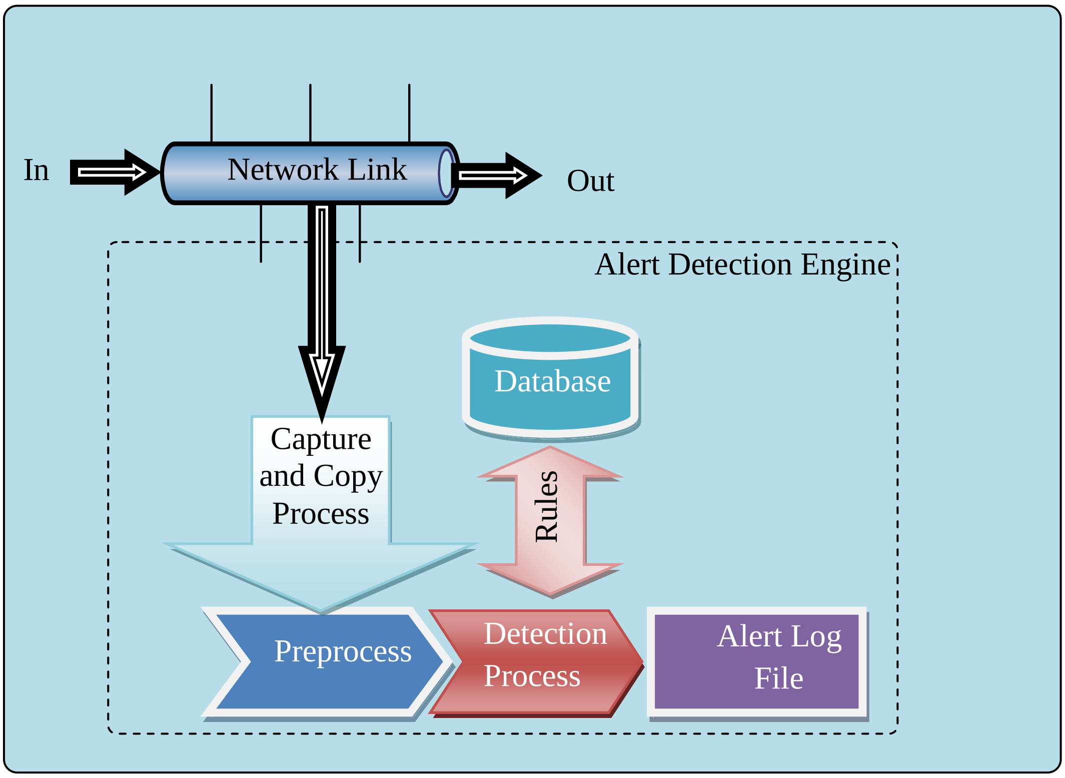

In
Network Link
Out
Alert Detection Engine
Database
Capture and Copy Process
Rules
Preprocess
Detection Process
Alert Log File

Figure1.1. Alert detection engine

A general diagram for the Alert Detection Engine is shown in Figure 1.1(Adnan, 2009). It consists of three segments:

1. **Capture and Copy Process**: this has the responsibility to capture and copy all incoming packets arriving from the network.

2. **Preprocess**: it is responsible for preparing captured packets for the detection engine. This includes classification of the packets according to protocols

3. **Detection process**: This process is responsible for checking the preprocessed packets for possible intrusions.

Alerts are any sort of user notification of an intruder activity. When detection engines detect an intruder, they will inform the security administrator. The alert information consists of several lines, representing the features of the alert (El-Taj, 2010). Most Alert Detection Engines logs contain certain common features. The details of these common features are as follows:

- ID: Unique identifier for the type of alert
- Date: Date of the occurrence
- Time: Time of the occurrence.
- IP Source: IP address of the attacker
- IP Destination: IP address of the victim.
- Port Source: Port source on the attacker

- Port Destination: Port destination on the victim.
- TTL: Time to live
- IpLen: The size of IP header in bytes.
- DgmLen: The size of packet in bytes.
- Protocol: The protocol used
- Priority: Priority of the alert.
- TOS: Type of Service
- CVE Ref.: Common vunerabilities and Exposures

1.4 Problem Statement

Despite the current capabilities of the Alert Detection Engine, it still suffers from the problem of generating a high level of false positive alerts. These can sometimes be as high as 99% (Elshoush & Osman, 2011; Tjhai, 2008). As a matter of fact, this is an important factor that determines the performance of the Alert Detection Engines. These high levels of false positive alerts have received the attention of researchers and has prompted them to seek different techniques to resolve the problem by reducing the rate of such false positives alerts. Alerts can also be better evaluated if certain methods are used to classify these threats based on their magnitude of threat.

The problem statement can summarized as follows:

1. There is a need to reduce the voluminous false alerts generated by Alert Detection Engines, which will increase the complexity and performance of the threat analysis system.

2. There is a need to assess each alert with a threat level. Even after reductions, some of the alerts may still be false or redundant alerts. Also, the alerts logged in the output of the Alert Detection Engines are not classified and are all trested as treats of equal magnitude. For the security analyst to better identify attacks and take appropriate remedial action against these threats, it would be useful to categorize the magnitude of each threat by attaching a threat level.

1.5 Research Objectives

The main objective of this thesis is to propose a framework which will consist of an enhanced false alert reduction system and an alert score assessment system. To achieve the above objective, the specific objectives are defined as follows:

1. To propose a new false alert reduction algorithm which will reduce false alerts based specifically on the Time feature and the Common Vulnerabilities and Exposures (CVE) feature.

2. To enhance two existing algorithms to support multiple features, which currently only support single feature, and add threat scores of features and alert to assess threat alert.

3. To evaluate the performance of proposed systems against existing Alert Detection Engines techniques.

1.6 Thesis Contributions

Alert Detection Engines analysts suffer from analyzing alerts because of the huge amount of redundant and false positive alerts. Their job is made even more difficult as there is no alert score assessment that would allow them to prioraties and assess threats based on threat level. This thesis provides solutions for these problems by providing the following:

a) **False Alert Reduction System*:*** This system is based on a new proposed Reduction Alert (NAR) Algorithm containing three phases. The first phase removes any redundant alert, based on the similarities of alerts features. The second phase removes redundant alerts based on the similarities of the alerts features with time threshold values. The third phase of this algorithm removes identified confirmed false positive alerts based on existing rules.

b) **Alert Score Assessment System:** This system contains of two sub-modules. The new proposed Generating Featuresets Sub-module is based on the Multi Feature Apriori (MFA) Algorithm. This algorithm is modified to analyse multiple features instead of its normal single feature analysis. The second sub-module is called the Generates Rules and Alert Score Sub-module. This sub-module is based on the Improve Rules Generation (IRG) Algorithm to generates rules and threat scores. This algorithm automatically calculates the threat scores of features and alerts.

c) **A False Alert Reduction and Alert Score Assessment Framework**: In this framework, the above two systems are integrated to form a complete system which can work on online alerts or offline alerts using alert log files. This system aims to make the Alert Detection Engines analyst’s task easier by reducing/removing some of the confirmed false positive alerts and also by providing them with new threat logs, in which, each threat has been assessed and marked with an appropriate threat level.

1.7 Research Methodology

Figure 1.2 illusetrate complete research methodology of this thesis.

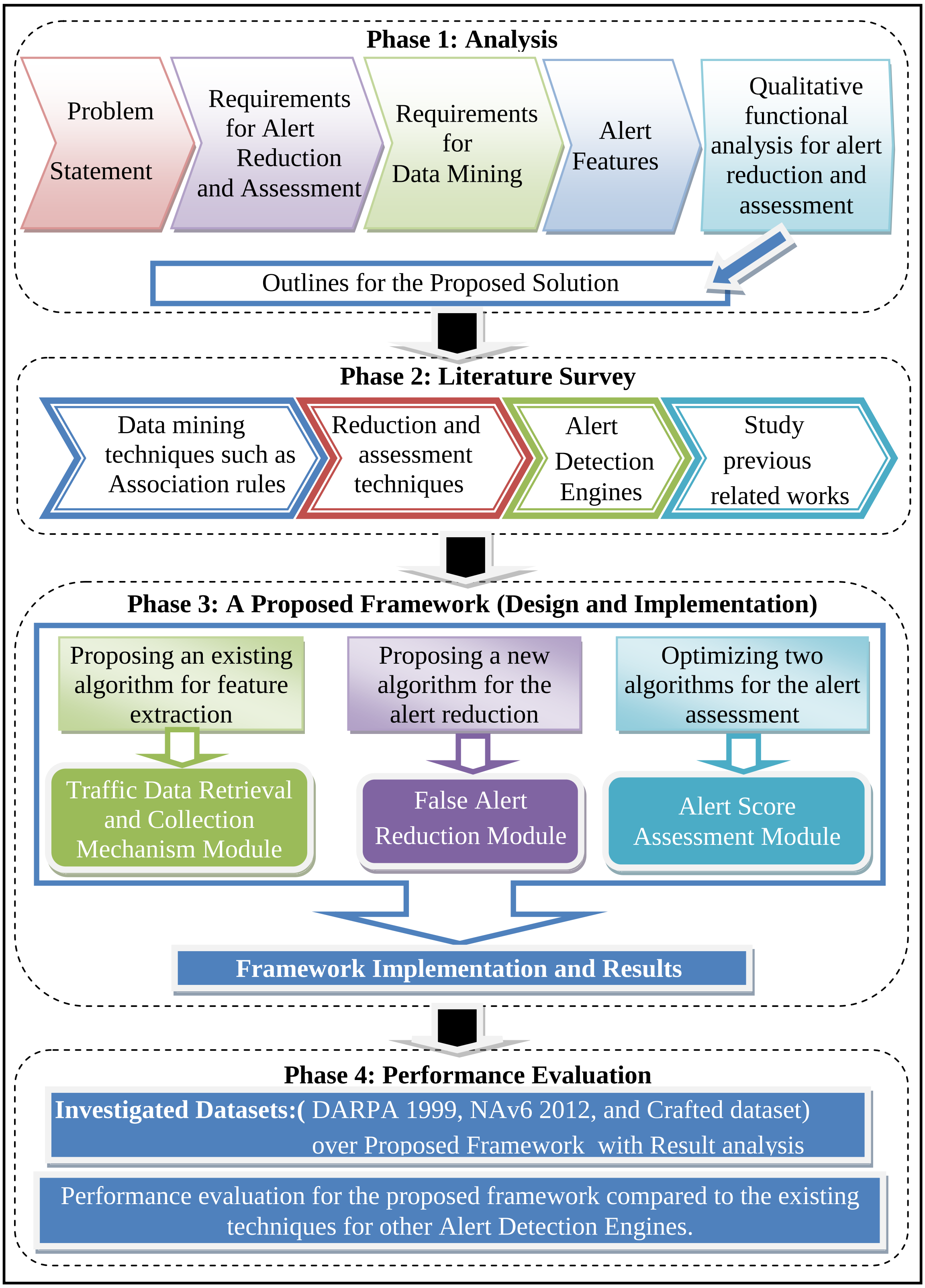

Phase 1: Analysis
Problem Statement
Requirements for Alert Reduction and Assessment
Requirements for Data Mining
Alert Features
Qualitative functional analysis for alert reduction and assessment
Outlines for the Proposed Solution
Phase 2: Literature Survey
Data mining techniques such as Association rules
Reduction and assessment techniques
Alert Detection Engines
Study previous related works
Phase 3: A Proposed Framework (Design and Implementation)
Proposing an existing algorithm for feature extraction
Proposing a new algorithm for the alert reduction
Optimizing two algorithms for the alert assessment
Traffic Data Retrieval and Collection Mechanism Module
False Alert Reduction Module
Alert Score Assessment Module
Framework Implementation and Results
Phase 4: Performance Evaluation
Investigated Datasets:(DARPA 1999, NAv6 2012, and Crafted dataset) over Proposed Framework with Result analysis
Performance evaluation for the proposed framework compared to the existing techniques for other Alert Detection Engines.

Figure 1.2. Research methodology

1.8 Thesis Outline

This thesis is organized into six chapters. The current chapter (Chapter One) presents an introduction and a brief background of the proposed research work.

Chapter Two introduces the current and related background studies in the areas of reducing and assessing false positive alerts using Alert Detection Engines.

Chapter Three presents the methodology of the proposed framework, and explains in detail the architectures of the framework.

Chapter Four describes the and presents the implementation details of the entire framework.

Chapter Five explains the experiments carried out to evaluate the proposed framework. It provides the evaluation results of each of the framework modules. It analyses and discusses the obtained results. It also includes comparisons against other related systems.

Chapter Six provides the conclusion of the work covered within this thesis and proposes directions for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter is divided into two main sections: Background and Related Work. The Background covers the Alert Detection Engines, Association rules for Data Mining, the Apriori algorithm and the Association rules generation algorithm are discussed.

The Related work covers; the alert reduction approaches, the alert classification techniques, the alert correlation techniques, as well as the alerts assessment. The comparison between other related approaches with the proposed approach are explained in the table at the end of this chapter.

2.2 Background

In the following sections, background subjects are highlighted with the Alert Detection Engine (ADE) and Data mining algorithms that are used in the proposed system.

2.2.1 Detection Engines

In recent years, research into Detection Engines have attracted a lot of work. Detection Engines checks the data that passes through networks and reveals any suspicious data. (Bace & Mell, 2011; Lars, 2008; Szmit et al., 2012; Tjhai, 2011).

A False Positive Alertisan alert that is released by the Detection Engine but it is actually not a threat. The Detection Engine triggers a huge amount of alerts. The average rate of false positive alerts from these alerts was up about 99% (Elshoush & Osman, 2011).

Security devices such as the firewalls, packet filtering, servers and IDS, work independently. They generate alerts which are written into a log file. The Detection Engine will then evaluate these log files and triggers alerts which will be sent to the analyst. The analyst will analyze these alerts to know the nature of the intrusion by using tools and techniques created for this purpose. This is to reduce the rate of the false positive alerts (Ignacio et al., 2008; Obbo, 2007; Tjaih, 2011). Nonetheless, there still exist weaknesses in processing such data due to the quantity of the data, even after the reduction process is done. A huge amount of false positive alerts still remain.

2.2.2 Alert Detection Engine Classification

The following methods are used to categorize the Alert Detection Engine:

- Information Source
- Type Analysis
- Response
- Detection time

Figure 2.1 shows the Alert Detection Engine classification (Elshoush & Osman, 2011). These methods are the most common standards that were adopted to classify the Alert Detection Engine.

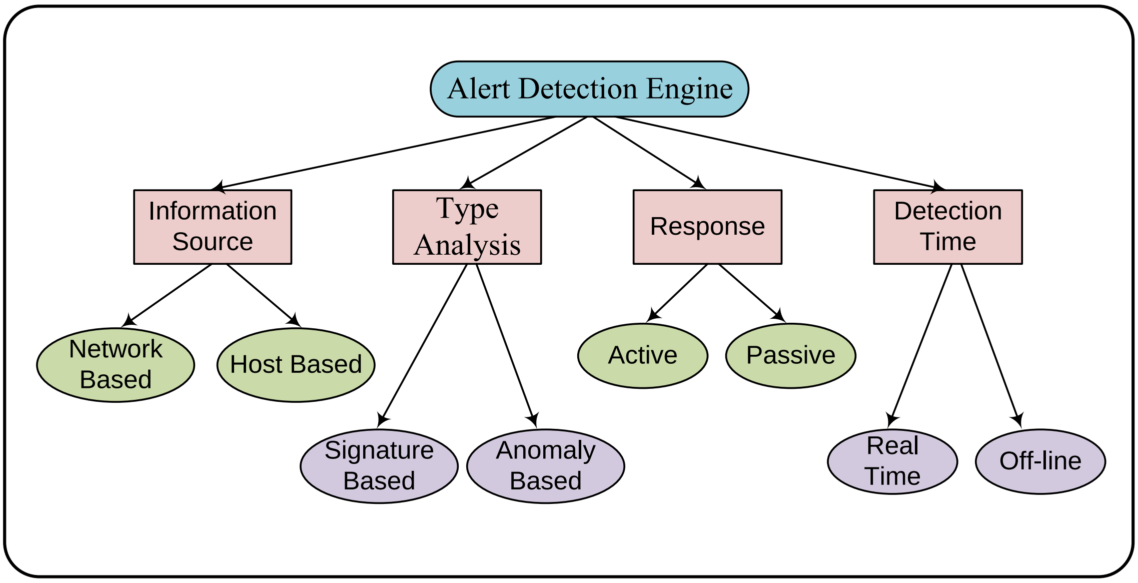

Alert Detection Engine
Information Source
Type Analysis
Response
Detection Time
Network Based
Host Based
Signature Based
Anomaly Based
Active
Passive
Real Time
Off-line

Figure 2.1. Classification of the alert detection engine

2.2.2.1 Information Source

The sources of the information may be classified into two ways. There are two types are (a) Network-Based Alert Detection Engine (NADE), and (b) Host-Based Alert Detection Engine (HADE).

These are generally some of the most important factors which are based on designing an Alert Detection Engine and are classified according to the location. Some Alert Detection Engines examine the network packets detected on the network backbone or LAN, where as others events which are triggered by the application of a software or when the operating system shows signs of intrusion (Lars, 2008; Szmit et al., 2012).

(a) Network-Based Alert Detection Engine (NADE)

This type of detection engine principally functions to detect intruder threats. It aims to copy all the packets which pass throughout the network portion which is being analysed and triggers an alert when a potential threat is detected. (Szmit et al., 2012).

NADEs can also consists of a group of sensors, which are hidden at different positions of the network. (Hoang et al., 2009). This form of detection engine, has strong flexibility and the possibility to be located at any point of the network. This is property which assists to increase security enhances the time to detect threats. (Pietro & Mancini, 2008; Xu & Ning, 2008). Unfortunately, at the same time, it has some disadvantages represented by its inability to detect if an attack is successful or unsuccessful. This is because it has a general configuration and triggers a high number of false positives.

(b) Host-Based Alert Detection Engine (HADE)

This type of detection engine works within a single host and is considered as the first point to detect intruders. Its work depends on the data being transferred to or from a computer. If the data originates from files in the operating system, it will analyze them and issue alerts when they are considered suspicious.

HADE has many advantages, some of which include the followings: it analyzes the data before it is encrypted; a feature which is not available in the NADE. It also controls events locally, as such detects threats which escape the ADE. Unfortunately, it also gives a large number of false positive alerts, in addition to being costly and time-consuming, since an HADE is required for each host. One important disadvantage of the HADE, however, is that it is easily disabled by the DDoS attacks (Xu & Ning, 2008).

2.2.2.2 Type Analysis

There are two methods of detecting and analyzing attacks : Anomaly-based Detection and Misuse-Based Detection. Anomaly-Based Detection is used in systems to detect suspicious activities and search for malicious patterns. Misuse-Based Detection is mainly used by commercial systems, and is based on suspicious behaviors (Wei-Zhou & Shun-Zheng, 2006).

(a) Anomaly-Based Detection

The function of the anomaly detector depends on suspicious behaviors in the network or the host. Based on the type of data being used, attacks behave differently from one case to another. Building a data file from normal behaviors of users is called a profile. Furthermore, its construction is based on the collected data through normal processes during the movement of this data within the network. In particular, this type of detection examines the data being used in different standards to categorize which of the events are considered normal and suspicious (Elshoush & Osman, 2011). Such systems that depend on this method detect any anonymous attacks. Nonetheless, the anomaly detector has a low accuracy, which results in the generation of a large number of false positive alerts (Szmit et al., 2012).

(b) Misuse-Based Detection

This method examines the system to search for activities which are identical to the signature or to a predetermined pattern describing an attack. A database is located within the system containing data about the patterns of the attacks. Therefore, the Misuse-based detector examines the network traffic depending on the information from the database, compared to the patterns that are obtained from the network, where a match sends out an alert.

Although these engines are effective in the detecting threats, they still generate some false positive alerts. Additionally, the engines are always required to update the databases with updated patterns of attacks, in order to obtain better results (Szmit et al., 2012; Wei-Zhou & Shun-Zheng, 2006). These engines are, however, more popular because they provide accurate alerts and could easily be developed.

One principal weakness of this method is its inability to identify non-predefined attacks which are not contained within the signature database. In any event, this knowledge is frequently and timely updated when there are new techniques of attacks or threats, even though it is difficult to fill all the knowledge bases (Jeremiah & Martin, 2011).

2.2.2.3 Response

The reactions of the ADE to analyze and detect an attack could always be grouped into **Passive Response** and **Active Response** categories. In general, the difference between the two classes is that the passive ADE sends reports to some other devices to take action, whereas the active ADE automatically begins reacting to such attacks (Morin et al., 2002).

(a) Passive Response

Using Passive Response, the system user or security manager is notified to fix what has occurred. The system administrator is notified via an alert message regarding the site and location of an attack. The system administrator then decides the next course of action to take, based on the information provided.

(b) Active Response

This method will initiate an automatic action to be taken in response to certain types of intrusions. These responses can be categorized as follows:

1) The ability to collect additional information: This involves increasing the sensitivity level of the sensor, in order to obtain additional clues of probable attacks. An example of this is the detection of all packages from the source at the beginning of the attack and during a specific period of time.

2) The ability to change the environment: This is a type of response that could stop an attack as is the case with a TCP connection. In this response, a session is closed when TCP and RST segments are introduced to the invader and the victim or when the IP address of the invader or the assaulted port is fitted to the access router or to the firewall, to avoid future attacks (Khalid, 2008).

2.2.2.4 Detection Time

Two major categories could be identified in the response detection time, which are; the “on-line” and “off-line” detections. When the intrusion detection process occurs in real time, it is categorized as on-line. When the process involves auditing data after it has been captures and stored, then it is termed as off-line.

Engines which combine both types of detection times on-and off-line are referred to as hybrid Engines (Elshoush & Osman, 2011).

2.2.3 Snort

Snort is an open-source ADE created by Roesch (Rafeeq, 2003). It is a very flexible and feasible software system that can be used with different types of databases, such as MySql, Oracle, and so on. This software has an attack-detection engine and a port scanner. It also helps warn or respond to any type of previously identified attacks and works well under many types of OS, such as Ms Dos, windows, and linux.

Snort provide two types of alerts: fast mode and full mode. The system presents options to the user to choose the required type of alert. Figure 2.2 shows the fast mode alert, whereas Figure 2.3 shows the full mode alert. This thesis uses the full mode alert because it contains most of the features that can be found in other types of ADE as well.

Alert1:

```
06/12-21:57:07.142376  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc
activity] [Priority: 3] {ICMP} 192.168.1.1 -> 192.168.1.5
```

Alert2:

```
06/12-21:57:21.284093  [**] [1:19669:1] WEB-CLIENT Telnet protocol specifier in
web page attempt [**] [Classification: Potential Corporate Privacy Violation]
[Priority: 1] {TCP} 172.16.114.50:80 -> 196.227.33.189:8013
```

Figure 2.2: Alert in fast mode

```
Alert1:
[**] [1:402:8] ICMP Destination Unreachable Port Unreachable [**]
[Classification: Misc activity] [Priority: 3]
08/07-20:08:31.452777 135.8.60.182 -> 172.16.112.100
ICMP TTL:63 TOS:0xC0 ID:1238 IpLen:20 DgmLen:106
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
172.16.112.100:137 -> 135.8.60.182:137
UDP TTL:127 TOS:0x0 ID:55809 IpLen:20 DgmLen:78
Len: 50  Csum: 24455 (50 more bytes of original packet)
** END OF DUMP
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2005-0068][Xref => http://cve.mitre.org/cgi-
bin/cvename.cgi?name=2004-0790]
Alert2:
 [**] [1:19669:1] WEB-CLIENT Telnet protocol specifier in web page attempt [**]
   [Classification: Potential Corporate Privacy Violation] [Priority: 1]
   06/12-21:57:21.284093 172.16.114.50:80 -> 196.227.33.189:8013
   TCP TTL:63 TOS:0x0 ID:5263 IpLen:20 DgmLen:1500
   ***A**** Seq: 0x9D677108  Ack: 0xC4275704  Win: 0x7C00  TcpLen: 20
[Xref => http://www.microsoft.com/technet/security/bulletin/MS11-064.mspx][Xref =>
http://cve.mitre.org/cgi-bin/cvename.cgi?name=2011-1965]
```

Figure 2.3: Alert in full mode

2.2.4 Alert log feature set

The alert log consists of a large volume of alerts that are issued from Alert Detection Engines. These alerts consists of a small section of real alerts and a large section of false alerts (Sundaram, 1996; Tadeusz, 2004).

Each alert log consists generally of several lines. Each of these lines contain several features that represent specific information about the alert. Using this information it is possible to identify between real and false alerts.

2.2.4.1 Common alert feature sets

Most Alert Detection Engines logs contain certain common features. When the alerts log comes from Alert Detection Engines, there is no standard on the format or what they should contain (Lars, 2008; Sundaram, 1996). However, all Alert Detection Engines logs contain certain standard common features including:

- **ID**, presents the unique identifier for the type of alert. This feature consists of three numbers.
    - The Generator ID number (GID) refers which component of ADE generated The alert.
    - Signature ID (SID), is written directly into the rule by using the SID option within ADE signatures.
    - Revision ID (RID) refers which revision of the signature was used.

- ***IPSource*** refers to the IP address of the host sending the packet. This IP address consist from 4 parts spreated by dots. It ranges from 0-255, for example: 172.16.112.207.

- ***Port Source*** refers to what port on the source it was sent from. It can range from 0- 65535**.**

- ***IP Destination*** refers to the IP address of the victim. Like ***IP Source***, it consist of 4 parts, exmaple: 202.126. 2.115.

- ***Port Destination*** refers to destination port on the victim. It is be as number such as 6667.

- ***Priority (Severity)*** it is often based on which classification it comes from, but can be overridden for each signature. It is a preset value determined by the signature. writer and is therefore a subjective value. it is devided to three level; High (1), Medium(2), and Low(3).

- **Date** refers to the date of attack happened, for Example 03/09.

- ***Time*** refers to time stamp of the time of occurrence. The Time feature consist from 4 parts (H:M:S.MS) the first represents the *H* for the hour, *M* for the minuts, *S* for the seconds, and *MS* for the micro seconds for example: *19:57:08.574994.*

- ***Protocol*** refers to the protocol used. It will in most cases be TCP, UDP, ICMP.

- ***TTL*** refers to time to live from the IP header in the packet. For example 32.

- ***IpLen*** refers to the size of IP header in bytes. For example 20 bytes.

- ***DgmLen*** refers to the size of packet in bytes. For example 50 bytes.

- ***CVE Ref***. it is refers to the website reference of CVE value.This is described in detail in section 2.2.4.3.

2.2.4.2 Time feature

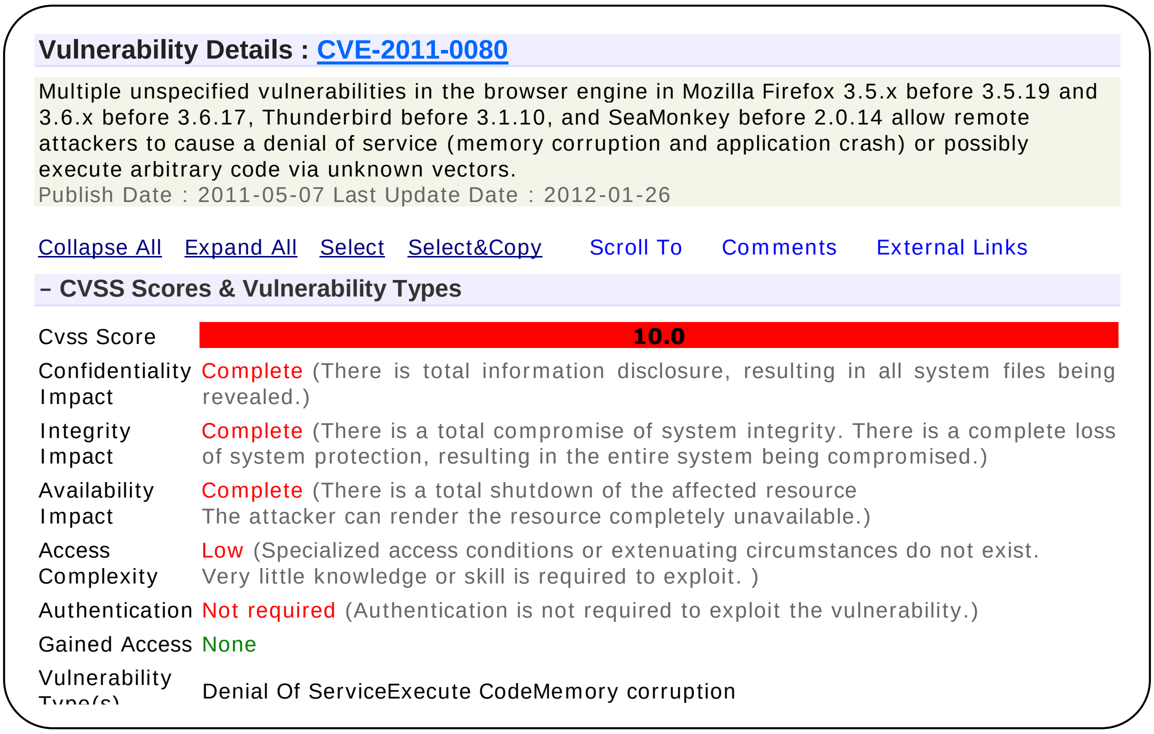
The time feature is a very significant feature because it states the time that an attack happened. This info can be used to determine the specific time used by an attacker to conduct a particular attack. In this case, redundant of alerts can be identified by comparing the time between alerts from the same source. In order to reduce the false alerts, the time feature can be considered as a more reliable method because there are several sensors are included in the ADE. As such, it can be detected when acknowledgement is released for the same threat from several sensors in extremely converged times (Elshoush & Osman, 2011). Using this information, the time feature with based on a threshold time value is used in this thesis.

2.2.4.3 Common Vulnerabilities and Exposures Feature (CVE)

CVE is the term used to refer to security threats and consists of two types, namely, **vulnerabilities** and **exposures**. Vulnerability refers to a computer, server, or network that is responsible for generating a definite and identifiable security risk in a particular context. Exposure refers to a security-related situation, event, or fact that might presented as a vulnerability to someone.

The MITRE Corporation (Mitre, 2012) developed the CVE to facilitate data-sharing process among diverse interests in security-related fields. CVE is a process of surfing for information using either security-related databases or the Internet. Such process is a collaboration of products from experts and representatives coming from different security-related organizations throughout the world.

Once the items in the CVE are generated, the items are given names based on two criteria: (1) the year of the formal inclusion of each item; and (2) the order of each item in the list for the given year. To illustrate further, consider CVE-2008-0080, which indicates a specific buffer overflow in the WebDAV Mini-Redirector in Microsoft Windows XP SP2, Server 2003 SP1 and SP2, and Vista. This condition may allow remote attackers to execute arbitrary codes via a crafted WebDAV response. The given item is thus added in 2008 and then given the sequence number 80 for that year (Mitre, 2012; Techtarget, 2011). Figure 2.4 shows an example of CVE information details.



Vulnerability Details : CVE-2011-0080

Multiple unspecified vulnerabilities in the browser engine in Mozilla Firefox 3.5.x before 3.5.19 and 3.6.x before 3.6.17, Thunderbird before 3.1.10, and SeaMonkey before 2.0.14 allow remote attackers to cause a denial of service (memory corruption and application crash) or possibly execute arbitrary code via unknown vectors.
Publish Date : 2011-05-07 Last Update Date : 2012-01-26

– CVSS Scores & Vulnerability Types

Cvss Score	10.0
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource The attacker can render the resource completely unavailable.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of ServiceExecute CodeMemory corruption

Figure 2.4. An example of CVE information details

In each alert, features that refer to the CVE reference are present and are relied upon in issuing the alert. When a return occurs to the CVE reference, the value of the score weakness or vulnerability is found. This value is then considered when calculating the threat score of the alert through its inclusion in the proposed research formula (Formula 4.1 in Chapter Four). The values of the CVE is contained within a system that is constantly updated and made available for all users.

2.2.5 Association Rules

Data mining is the process of extraction and analysis, by automatic or semiautomatic means, of large quantities of data in order to discover meaningful patterns and rules (Berry et al., 2000; Chen et al., 2010; Cios et al., 2010; Elis, 2001; Shilpa & Sunita, 2011). Data mining tools uncover hidden information. Relationship between variables and customer behavior that are non-intuitive are the gems that data mining hopes to figure out. There are different applications of data mining including in the business area, customer segmentation, market basket analysis, risk management, fraud detection, delinquency tracking, and demand prediction. The principal goals of data mining in practice can be classified into two categories, **Description** and **Prediction** (Cios et al., 2010; Kantardzic, 2011; Mamdouh, 2009; Tjha et al., 2008).

Description is the process of trying to find relationships and patterns within the dataset .Patterns that are found are described in the form of rules based on statistical evidence.

Prediction involves variables or fields within the database to predict unknown or future values of other variables of interest. In other words, prediction is the process of using the training data to predict the future value of a feature based on the values of other features. **Prediction** technique is common used in neural networks, classification, and decision trees to find the relations between the attribute values (Al-Shalabi, 2011; Goethals, 2003; Klaus & Marc, 2002).

In general the data mining task involves; classification, clustering, similarity sequence discovery, sequential patterns, and association rules. This thesis focuses on association rules, because of the advantage of the benefits of association rules. This is important in finding a relationship between correlation confidence elements, through the generation of set of items that are on that basis the extraction roller and through which the degree of correlation confidence between the elements is calculated.