

**IMPROVING THE THREAT DETECTION
PERFORMANCE OF A NETWORK INTRUSION
DETECTION SYSTEM USING A 3-TIER FRAMEWORK**

Samira Dinusha Wickrama Senanayake

(179350H)

Dissertation submitted in partial fulfilment of the requirements for the degree
Master of Science in Computer Science specializing in Security Engineering.

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2021

DECLARATION OF THE CANDIDATE & SUPERVISOR

I declare that this is my own work, and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: ***UOM Verified Signature***

Date: 24-03-2021

The above candidate has carried out research for the Masters Dissertation under my supervision.

Name of the supervisor: Dr C.D. Gamage

Signature of the supervisor:

Date: 24-03-2021

Abstract

Information security is becoming more and more critical for data and information. Network security plays a major role in securing the data and systems from Cyber adversaries. It is crucial to detect the dangers actively and implement defences to protect network infrastructure from Cyber-attackers. In this project, we have introduced a way to optimise the threat detection capabilities using Zeek Network Security Monitor and Weka machine learning application. In fact, we have performed a comprehensive study on the evolution of Intrusion Detection Systems (IDS) using the past literature and identified the factors that contributed to both improved performance and limitations in threat detection. We have designed and developed a Network Security Monitoring (NSM) system prototype using Zeek NSM, Elasticsearch, Filebeat and Kibana Stack(EFK stack) and Weka application.

Moreover, our prototype actively performs network surveillance and alerts the user in an event of intrusion. Finally, we have performed a passive machine learning analysis using Random Forrest, K-Nearest Neighbors and Naïve Bayes classifiers on Denial of Service, Reconnaissance and Worm attacks. We have used a sample set of data from the UNSW-NB15 data set for the machine learning analysis activities.

Installation and configuration of open-source applications are not always straightforward, and they could be swamped with cumbersome processes. We have provided foolproof, stepwise guidance to perform the installation and configure of the Zeek and EFK stack at the end of this thesis.

The authors main objective is to design and develop user-friendly security solutions for threat detection using open-source applications. This project is the initial step to achieve that objective.

Keywords: Network Security, NIDS, Zeek NSM, Weka

DEDICATION

I dedicate this work to all my teachers, lecturers and mentors who taught me not to give up when life is challenging and encourage me to be at my best. May they live a long and healthy life!

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who contributed to carry out this study and make this project a success.

Firstly, I would like to thank my research supervisor Dr C.D. Gamage for giving me valuable advice, guidance, and suggestions to carry out this study.

Secondly, I would like to thank my parents, Ms Vindya Munaweera, Mr Duleep Nakandala, Mr Damsenevi Gurudeniya and my MSc. batchmate Mr Pansilu Pitigalaarachchi who encouraged me and supported me in numerous ways for the whole duration of this project to make this study a reality.

Next, I would like to thank my office colleagues at HSBC Data Processing Lanka (Pvt.) Ltd for all the support and advice given to me to accomplish this project. I would like to mention Mr Kasun Attanayake, Mr Chathura Yatawatte, Mr Lasantha Dasanayaka, Mr Yashodharan Sinnathamby, Mr Hamada Packeer, Mr Richard Baines, Mr Gavin Hawkins, and Mr Colin Fawkes for supporting me in my higher educational activities.

Finally, I would like to thank the Zeek open-source community for being the helping-hand when I needed it the most. I should mention and thank Dr Vern Paxson, Mr Yacin Nadji and Mr Pierre Gaulon for all the advice they provided me through the virtual space.

TABLE OF CONTENTS

DECLARATION OF THE CANDIDATE & SUPERVISOR	i
Abstract	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	x
LIST OF ABBREVIATIONS	xii
LIST OF APPENDICES	xiii
1. INTRODUCTION	1
1.1 Motivation	1
1.2 Security in Computer Networks	2
1.2.1 The Internet	2
1.2.2 The World Wide Web (WWW)	2
1.2.3 Information Security	3
1.2.4 Cybersecurity	3
1.2.5 Detecting Anomalies in Computer Networks	4
1.2.6 Introduction to Intrusion Detection Systems (IDS)	7
1.2.7 Open-source NSM Tools and Their Capabilities	13
1.2.8 Zeek Network Monitoring System	14
1.3 Research Gap	14
1.4 Problem Statement	15
1.5 Aspirations of the Authors	15
1.6 Research Question	16
1.7 Report Organisation	16
2. LITERATURE REVIEW	17
2.1 Intrusion Detection System's Role in Securing the Computer Networks	17
2.1.1 TCP/IP Model	18
2.1.2 Intrusion Detection Systems (IDS)	23
2.1.3 Threat Detection Architectures	28
2.1.4 Threat Detection Metrics for IDSs	31
2.2 Cybersecurity Threats and Threat Detection	36
2.2.1 Anomaly Detection Techniques	39

2.2.2 Machine Learning and Intrusion Detection	45
2.2.3 Feature Selection for Intrusion Detection	53
2.2.4 Significant Challenges in Anomaly Detection	55
2.3 Zeek Network Monitoring System	57
2.3.1 Zeek Architecture	60
2.3.2 Zeek Log Files	61
2.4 Summary	63
3. METHODOLOGY	64
3.1 Overview of Proposed Approach	65
3.1.1 Capture and Analyse Network Traffic in the Test Environment	66
3.1.2 Extract and Pre-process Zeek log Files for Machine Learning	66
3.1.3 Perform Passive Machine Learning Analysis with Weka	66
3.2 Tools and Environments	67
3.2.1 Elasticsearch, Logstash and Kibana (ELK)	67
3.2.2 Weka	68
3.2.3 Network-based Data Sets	69
3.2.4 Test-bed Specifications	76
3.3 The Proposed NIDS Solution	77
3.4 Summary	78
4. IMPLEMENTATION	79
4.1 Capture and Analyse Network Traffic in the Test Environment	80
4.2 Extract and Pre-process Zeek log Files for Machine Learning	84
4.2.1 Data Pre-Processing Procedure	84
4.2.2 Feed Data Files to Weka Application	88
4.3 Perform Passive Machine Learning Analysis with Weka	88
4.4 Training ML Models Using a Sample Set from the UNSW-NB15 Data Set	95
4.4.1 Training the ML Model	101
4.5 Summary	103
5. DISCUSSION AND CONCLUSIONS	104
5.1 Findings	104
5.1.1 Discussion on the Random Forrest Classifier Training	105
5.1.2 Discussion on KNN Classifier Training	106
5.1.3 Discussion on Naïve Bayes Classifier Training	107
5.1.4 Discussion on Attribute Ranking	109
5.2 Challenges and Limitations	110

5.2.1 Installation and Configuration Process	110
5.2.2 EFK? Why Not ELK	111
5.2.3 Storage Issues	111
5.2.4 Integrating Machine Learning to Optimise the Detection Rate	111
5.2.5 Insufficient Data to Train Machine Learning Model – Worm Attacks	112
5.2.6 Log File Analysis	112
5.2.7 Real-Time Threat Detection Using Machine Learning	112
5.3 Conclusion	113
5.4 Future Work	113
REFERENCE LIST	115
APPENDICES	122
Appendix A: Zeek Installation & Configuration	122
Appendix B: Elasticsearch Installation & Configuration	128
Appendix C: Filebeat Installation & Configuration	130
Appendix D: Kibana Installation & Configuration	131
Appendix E: Zeek-EFL Integration	132
Appendix F: Weka Installation	139
Appendix G: Data Pre-processing Process	140
Appendix H: Weka – Classifier Outputs	141
Appendix I: Weka – Ranking the Attributes (Features)	150
Appendix J: Weka – Decision Table Predictive Model Output Information	152

LIST OF FIGURES

Figure 1.1	Intrusion Detection System Example	8
Figure 1.2	Host-based Intrusion Detection System Example	9
Figure 1.3	Network-based Intrusion Detection System Example	10
Figure 1.4	Timeline of Zeek's History [65]	15
Figure 2.1	TCP/IP Reference Model	19
Figure 2.2	TCP Header Format	21
Figure 2.3	IP Header Format	22
Figure 2.4	Firewall Example	23
Figure 2.5	Intrusion Prevention System Example	25
Figure 2.6	Three Major Themes for Improving SOC Operations [47]	26
Figure 2.7	IDS Alert Categorisation	32
Figure 2.8	Percentage Compromised by at Least One Successful Attack, by Year	37
Figure 2.9	Number of Cyber Security Incidents Report to SL CERT from	37
Figure 2.10	Growth of the Types of Cybersecurity Incidents in Sri Lanka	38
Figure 2.11	Signature-based Detection	39
Figure 2.12	Anomaly-based Detection	40
Figure 2.13	Classification of Different Learning	46
Figure 2.14	Common Machine Learning Algorithms	47
Figure 2.15	Integrating Machine Learning to a NIDS	51
Figure 2.16	The Number of Papers Published on ML and Cybersecurity from 2010-2018	52
Figure 2.17	Zeek's Internal Architecture	60
Figure 3.1	Test-bed	64
Figure 3.2	Overview of Proposed Approach	65
Figure 3.3	High-level Architecture to Improve Threat Detection Capabilities	77
Figure 4.1	Implementation Plan	79
Figure 4.2	Online Network Traffic Analysis	81
Figure 4.3	Zeek Offline Network Traffic Log Generation	82
Figure 4.4	Offline Network Traffic Analysis by Replaying .pcap file	83
Figure 4.5	Data Pre-Processing Procedure	84
Figure 4.6	Use Case - Create scan_netflow.pcap file	85
Figure 4.7	Use Case - Generate Conn.log using Zeek utilities	86
Figure 4.8	Malicious.csv file	86
Figure 4.9	Use Case - Generate Conn.log for smallFlows.pcap file	86
Figure 4.10	Normal.csv file	87
Figure 4.11	trainset.arff file format	87
Figure 4.12	testset.arff file format	87
Figure 4.13	Feed .arff files to Weka Application (Training Data and Testing Data)	88
Figure 4.14	Weka Explorer GUI	89
Figure 4.15	Results of the Decision Table Algorithm	89
Figure 4.16	Results of the J48 Algorithm	90
Figure 4.17	Decision tree Reliance on the Time Feature	91
Figure 4.18	Results of the J48 Algorithm After the Re-configuration	91
Figure 4.19	Unprocessed testset.arff	92
Figure 4.20	ML Model Predictions	93
Figure 4.21	First 10 instances of test.csv file and the output of predictive model	93

Figure 4.22 Visualising the Predicted Labels for the Test Data Set	94
Figure 4.23 Procedure to Test the Selected Sample Data	98
Figure 4.24 UNSW-NB15 Sample Data Set Testing Framework	99
Figure 4.25 Upload the Master.csv file to Weka	100
Figure 4.26 Relationship Between the Attributes (Features) and the Classes	100
Figure 4.27 Result of the ML Analysis – Random Forrest Classifier	101
Figure 4.28 Result of the ML Analysis – KNN Classifier	102
Figure 4.29 Result of the ML Analysis – Naïve Bayes Classifier	103
Figure 5.1 Weka Interface - Current Relation & Selected Attribute	104
Figure 5.2 Classifier Performance Comparison	108
Figure 7.1 VirtualBox Host-Only Ethernet Adapter Setup	122
Figure 7.2 Setup the Sniffing Interface	123
Figure 7.3 Kibana Configuration - Kibana.yml	132
Figure 7.4 Elasticsearch Configuration - Elasticsearch.yml	133
Figure 7.5 Filebeat Configuration ii - Filebeat.yml	134
Figure 7.6 Filebeat Configuration i - Filebeat.yml	134
Figure 7.7 Filebeat Configuration iii - Filebeat.yml	135
Figure 7.8 zeek.yml File Configuration	136
Figure 7.9 Checking the application status	137
Figure 7.10 Accessing Kibana	138
Figure 7.11 Weka GUI	139

LIST OF TABLES

Table 1.1	Most Common Network Attacks	6
Table 1.2	Comparative Study of Well-known Open-source IDSs	13
Table 2.1	TCP/IP Layers [23]	19
Table 2.2	Layer-specific Protocols in TCP/IP Reference Models [40]	20
Table 2.3	Difference Between IDS and Firewalls	24
Table 2.4	IDS vs. IPS in Nutshell	25
Table 2.5	Categorisation of Intrusion Detection Systems	27
Table 2.6	Comparison of HIDS and NIDS Technologies	29
Table 2.7	Advantageous and Limitations of the Statistical Based Approach	41
Table 2.8	Advantageous and Limitations of the Clustering Approach	41
Table 2.9	Advantageous and Limitations of the Finite State Machine Approach	42
Table 2.10	Advantageous and Limitations of the Classification Based Approach	42
Table 2.11	Advantageous and Limitations of the Information Theoretic-based Approach	43
Table 2.12	Advantageous and Limitations of the Evolutionary Computation Approach	43
Table 2.13	Advantageous and Limitations of the Hybrid Anomaly Detection Approach	44
Table 2.14	Comparison of Intrusion Detection Techniques	44
Table 2.15	Characteristics of SIDS and AIDS	44
Table 2.16	Pros and Cons of ANN	47
Table 2.17	Pros and Cons of SVM	48
Table 2.18	Pros and Cons of KNN	49
Table 2.19	Pros and Cons of Naive Bayes	49
Table 2.20	Pros and Cons of LR	49
Table 2.21	Pros and Cons of Decision Tree	50
Table 2.22	Pros and Cons of K-means	50
Table 2.23	Malware Behavioural Features Collated by Akinrolabu [44]	54
Table 2.24	Identified Features by the Interviews (Akinrolabu) [44]	55
Table 2.25	Logs Related to Network Protocols	62
Table 2.26	Logs Related to Threat Detection	62
Table 2.27	Logs Related to Network Observation	63
Table 3.1	Properties of UNSW-NB15 Data Set	71
Table 3.2	UNSW-NB15 Record Distribution	72
Table 3.3	Features of UNSW-NB15	74
Table 3.4	Test-bed Virtual Machines	76
Table 3.5	Applications in Use	77
Table 4.1	Use Case - Online Network Traffic Analysis	80
Table 4.2	Use Case - Offline Network Traffic Analysis	82
Table 4.3	Use Case - Offline Network Traffic Analysis by Replaying .pcap file	82
Table 4.4	Use Case - Creating .arff Files Using Conn.logs	85
Table 4.5	Confusion Matrix - Decision Table Algorithm	90
Table 4.6	Confusion Matrix - J48 algorithm	91
Table 4.7	Confusion Matrix - J48 algorithm After Re-configuration	92
Table 4.8	Confusion Matrix - Decision Table Predictive Model	94
Table 4.9	UNSW-NB15 No. Records on the Selected Attacks	96
Table 4.10	UNSW-NB15 Features Proposed by ARM Algorithm	97
Table 5.1	Detailed Accuracy by Class - Random Forrest	105

Table 5.2	Confusion Matrix - Random Forrest	105
Table 5.3	Detailed Accuracy by Class - KNN	106
Table 5.4	Confusion Matrix - KNN	106
Table 5.5	Detailed Accuracy by Class - Naïve Bayes	107
Table 5.6	Confusion Matrix - Naïve Bayes	107
Table 5.7	Ranked Attributes Using Weka	109

LIST OF ABBREVIATIONS

DoS	Denial-of-service
EFK	Elasticsearch, Filebeat, Kibana
ELK	Elasticsearch, Logstash, Kibana
HIDS	Host-based Intrusion Detection System
IDS	Intrusion Detection System
IP	Internet Protocol
KNN	K-Nearest Neighbor Algorithm
ML	Machine Learning
NIDS	Network-based Intrusion Detection System
NSM	Network Security Monitor
TCP	Transmission Control Protocol
TSV	Tab-separated Values
WEKA	Waikato Environment for Knowledge Analysis
.arff	Attribute-Relation File Format
.csv	Comma-separated values Format
.pcap	Packet Capture Format

LIST OF APPENDICES

Appendix – A	Zeek Installation & Configuration	122
Appendix – B	Elasticsearch Installation & Configuration	128
Appendix – C	Filebeat Installation & Configuration	130
Appendix – D	Kibana Installation & Configuration	131
Appendix – E	Zeek-EFL Integration	132
Appendix – F	Weka Installation	139
Appendix – G	Data Pre-processing Process	140
Appendix – H	Weka – Classifier Outputs	141
Appendix – I	Weka – Ranking the Attributes (Features)	150
Appendix – J	Weka – Decision Table Predictive Model Output	152