The International Journal of Multi-Disciplinary Research
ISSN: 3471-7102, ISBN: 978-9982-70-318-5

Use of Network Intrusion Detection System on School Networks

(Conference ID: CFP/835/2018)

By: Lawrence Tembo
tigerlawr@yahoo.co.uk
Dept of ICTs
School of Engineering
Information and Communications
University

Advisor: Dr. R. Silumbe
Dept of ICTs
School of Engineering
Information and Communications
University

ABSTRACT

The goal of a network-based intrusion detection system (IDS) is to identify malicious behavior that targets a network and its resources. Intrusion detection parameters are numerous and in many cases they present uncertain and imprecise causal relationships which can affect attack types. A Bayesian Network (BN) is known as graphical modeling tool used to model decision problems containing uncertainty. In this paper, a BN is used to build automatic intrusion detection system based on signature recognition. The goal is to recognize signatures of known attacks, match the observed behavior with those known signatures, and signal intrusion when there is a match. A major difficulty of this system is that intrusions signatures change over the time and the system must be retrained. An IDS must be able to adapt to these changes. The goal of this paper is to provide a framework for an adaptive intrusion detection system that uses Bayesian network.

Malicious behavior is defined as a system or individual action which tries to use or access to computer system without authorization (i.e., crackers) and the privilege excess of those who have legitimate access to the system (i.e., the insider threat).

The proliferation of heterogeneous computer networks has serious implications for the intrusion detection problem. Foremost among these implications is the increased opportunity for unauthorized access that is provided by the network's connectivity.

Bayesian techniques create a plan of goal-directed actions. An event classification scheme is proposed based on Bayesian networks. Bayesian networks improve the aggregation of different model outputs and allow one to seamlessly incorporate additional information.

Although a wide range of security technologies such as information encryption, access control, and intrusion prevention can protect network-based systems, there are still many undetected intrusions. For example, firewalls cannot prevent internal attacks. Thus, Intrusion Detection Systems (IDSs) play a vital role in network security.

ACKNOWLEDGEMENTS

Firstly, I would like to express many thanks to my supervisor Dr. Richard Silamba, for his fully-valued advices, useful comments and feedback.
I would like to thank my family and friends who patiently supported, given me the strength and courage in the past three years to work on this project. Without their support and words of encouragement I would have never achieved this goal.

Lawrence Tembo

Table of Contents

Chapter 1.

Introduction

A single intrusion of a computer network can result in the loss or unauthorized utilization or modification of large amounts of data and causes users to question the reliability of all of the information on the network. There are numerous methods of responding to a network intrusion, but they all require the accurate and timely identification of the attack [1, 2].

Intrusion detection system (IDS) is an important component to maintain network security. Also, as most schools and government institutions in Zambia are quickly embracing technology and going online, it is useful and necessary to build an effective IDS for the network. Malicious behavior is defined as a system or individual action which tries to use or access a computer system without authorization (i.e., crackers) and the privilege excess of those who have legitimate access to the system (i.e., the insider threat).

Security policies or firewalls have difficulty in preventing such attacks because of the hidden weaknesses and bugs contained in software applications.
Moreover, hackers constantly invent new attacks and disseminate them over the internet. Disgruntled employees, bribery and coercion also make networks vulnerable to attacks from the inside. Mere dependence on the stringent rules set by security personnel is not sufficient. Intrusion detection systems (IDS), which can detect, identify and respond to unauthorized or abnormal activities, have the potential to mitigate or prevent such attacks [3].

This paper proposes a network based IDS which is a distributed system with an adaptive architecture so as to make full use of the available resources without overloading any single machine on the network.

The proposed IDS can detect new types of attacks with fairly accurate results. Evaluation of the proposed IDS on a network shows that it is a promising approach to detecting attacks on the network infrastructure.

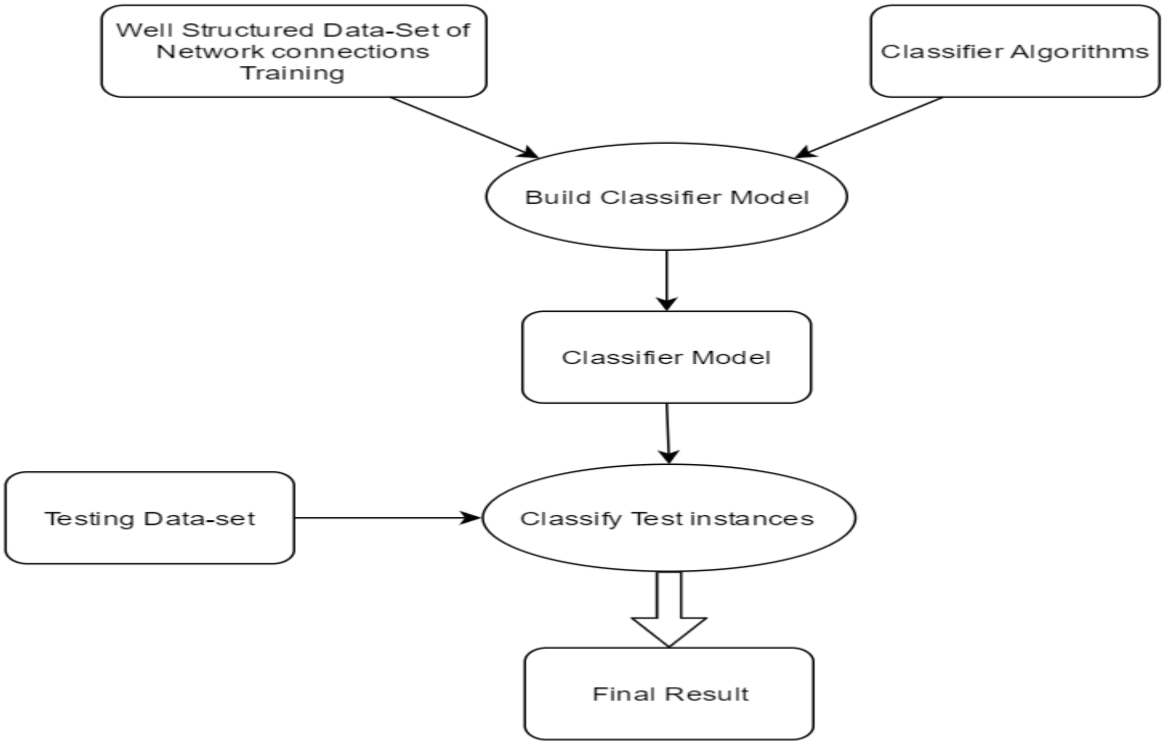

Figure 1. Approach

1.1 Objectives of the project

The following points explain the objectives of this project.

a) To develop intrusion detection system using agents.

b) To protect secure information of the school from outside and inside intruders

c) To decrease the bottleneck in the main server by distributing the sensors in the particular hosts.

d) To provide a secure environment for work.

e) The log of the intrusion detection system can be used for forensic purpose to find out the culprit.

f) The goal is to recognize signatures of known attacks, match the observed behavior with those known signatures, and signal intrusion when there is a match.

1.2 Purpose

Using networks to share data is known by more and more people due to its advantages such as high scalability and high flexibility. Network service users usually do not need to know how the network based software or platform runs; instead, they only need to send the requests to the network and then wait for the results, which is a much easier and more efficient way to access the needed computing resources. However, there are several issues for the current network platforms. security issues such as information leakage, unreliable data and unauthorized access are the most concerned problems by the majority of users. Other issues such as stable operations, support systems and user friendliness have received less attention.

1.3 Problem Statement

The Need for a Network Intrusion System

These days' information is the most valuable and powerful tool. This information is also becoming vulnerable to the hackers, intruder, etc. who can do damage to the network. That's why it's important for most institutions to set up a Network Intrusion Detection System.

The system can be used in the following:

a) **Banks and Financial Institutions:** The system will certainly minimize the unauthorized access and take immediate response to stop such illegal access.

b) **Police Department:** They have very valuable and sensitive information. This information should be kept secretly. So, the computer networks and terminals should have intrusion detection system for the safety of the information against the probable intruders.

c) **Government:** Recently almost all governments are going digital hence there need to secure the networks transmitting the information and data NIDS can help mitigate and protect the network for unauthorized access.

d) **Universities, Educational Sector:** Most of the examinations are done in digitally transferable medium. The election voting is also done and counted through computer medium. All this information is to be secured and must be alerted with possible intruders.

Chapter 2.

Background and review of Literature

A communications network forms the backbone of any successful organization. These networks transport a multitude of applications and data, including high-quality video and delay-sensitive data such as real-time voice. The bandwidth-intensive applications stretch network capabilities and resources and complement, add value, and enhance every business process. Networks must provide secure, predictable, measurable, and sometimes guaranteed services. Achieving the required quality of service (QoS) by managing delay, delay variation (jitter), bandwidth, and packet loss parameters on a network is the answer to a successful end-to-end business solution.

With the tremendous growth of network-based services and sensitive information on networks, network security is getting more importance than ever. Security is a big issue for all networks in today's enterprise environment. Hackers and intruders have made many successful attempts to bring down high-profile company networks and web services. Many methods have been developed to secure the network infrastructure and communication over the Internet, among them the use of firewalls, encryption, and virtual private networks. Intrusion detection is a relatively new addition to such techniques. Intrusion detection methods started appearing in the last few years. Using intrusion detection methods, we can collect and use information from known types of attacks and find out if someone is trying to attack your network or particular hosts.

2.1 Background

The number of information warfare attacks is increasing and becoming increasingly sophisticated. Annual reports indicate a significant increase in the number of computer security incidents each year. Not only are these attacks becoming more numerous, they are also becoming more sophisticated. Each attacked computer has limited information on who is initiating the attack and from where. The threat of a sophisticated computer attacks is growing. Unfortunately, intrusion detection and response systems have not kept up with the increasing threat.

There are two general categories of attacks which intrusion detection technologies attempt to identify - anomaly detection and misuse detection, refer to Fig. 1. Anomaly detection identifies activities that vary from established patterns for users, or groups of users. Anomaly detection typically involves the creation of knowledge bases that contain the profiles of the monitored activities. The second general approach to intrusion detection is misuse detection. This approach involves the comparison of a user's activities with the known behaviors of attackers attempting to penetrate a system. While anomaly detection typically utilizes threshold monitoring to indicate when a certain established metric has been reached, misuse detection approach frequently utilize a rule-based approach.

When applied to misuse detection, the rules become scenarios for network attacks. The intrusion detection mechanism identifies a potential attack if a user's activities are found to be consistent with the established rules. The use of comprehensive rules is critical in the application of expert systems for intrusion detection [1].

Figure. 2 The Flow Chart of Misuse Detection and Anomaly Detection Application [10].

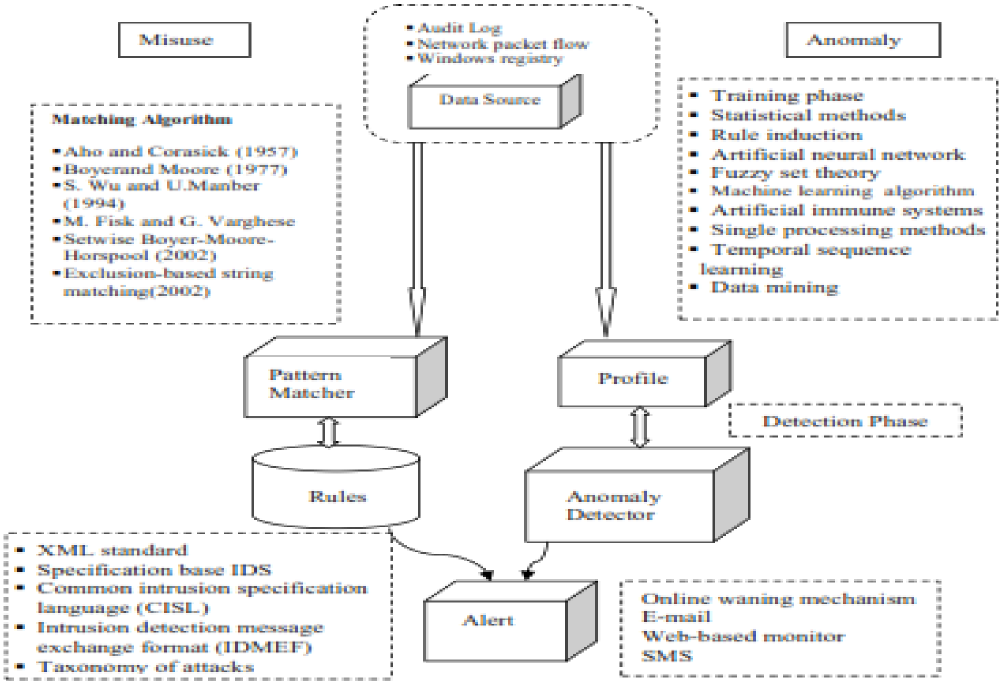
Misuse
• Audit Log
• Network packet flow
• Windows registry
Data Source
Anomaly
Matching Algorithm
• Aho and Corasick (1957)
• Boyerand Moore (1977)
• S. Wu and U.Manber (1994)
• M. Fisk and G. Varghese
• Setwise Boyer-Moore-Horspool (2002)
• Exclusion-based string matching(2002)
• Training phase
• Statistical methods
• Rule induction
• Artificial neural network
• Fuzzy set theory
• Machine learning algorithm
• Artificial immune systems
• Single processing methods
• Temporal sequence learning
• Data mining
Pattern Matcher
Profile
Detection Phase
Rules
Anomaly Detector
• XML standard
• Specification base IDS
• Common intrusion specification language (CISL)
• Intrusion detection message exchange format (IDMEF)
• Taxonomy of attacks
Alert
Online waning mechanism
E-mail
Web-based monitor
SMS

Figure. 2

2.2 Access Control

Access control is concerned with determining the allowed activities of legitimate users, mediating every attempt by a user to access a resource in the system. A given information technology (IT) infrastructure can implement access control systems in many places and at different levels. Networks use access control to protect access to the files and resources.

The main objective of IT is to make information available to users and applications. A greater degree of sharing may get in the way of resource protection; in reality, a well-managed and effective access control system actually facilitates sharing. A sufficiently fine-grained access control mechanism can enable selective sharing of information where in its absence, sharing may be considered too risky altogether.

In general, access control mechanisms require that security attributes be kept for users and resources. User security attributes can consist of categories such as user identifiers, groups, and roles to which users belong, or they can include security labels reflecting the level of trust bestowed on the user. Resource attributes can take on a wide variety of forms. For example, they can consist of sensitivity labels, types, or access control lists. In determining a user's ability to perform operations on a resource, access control mechanisms compare the user's security attributes to those of the resource.

2.3 Telecommunications and Network security

The Telecommunications and Network Security domain includes the structures, transmission methods, transport formats, and security measures that provide confidentiality, integrity, availability, and authentication for transmissions over private and public communications networks and media. This domain is the information security domain that is concerned with protecting data, voice, and video communications and ensuring the following:

The fundamental information systems security concept of C.I.A. relates to the Telecommunications domain in the following three ways.

Confidentiality

Confidentiality is the prevention of the intentional or unintentional unauthorized disclosure of contents. Loss of confidentiality can occur in many ways. For example, loss of confidentiality can occur through the intentional release of private company information or through a misapplication of network rights.

Integrity

Integrity is the guarantee that the message sent is the message received and that the message is not intentionally or unintentionally altered. Loss of integrity can occur either through an intentional attack to change information (for example, a web site defacement) or, most commonly, through accidental alteration of data by an operator.

Availability

This concept refers to the elements that create reliability and stability in net-works and systems. Availability ensures that connectivity is accessible when needed, allowing authorized users to access the network or systems.

The use of ill-structured security mechanisms can also affect availability. Over engineered or poorly designed security systems can impact the performance of a network or system as seriously as an intentional attack can.

2.4 TCP

TCP is a reliable connection-oriented protocol used with well-known applications such as telnet or smtp. An application such as telnet cannot tolerate the uncertainty of the Internet Protocol that can lose datagrams or deliver them in a different order from which they were sent. TCP is the protocol that orchestrates and ensures reliability. It does so using the following mechanisms:

- **Exclusive TCP connection.** When a TCP session is established, the connection is exclusive and unique between the two hosts. This kind of connection is called a unicast connection. The negotiation of the unique session allows both sides to track the traffic exchanged between the two hosts.

- **TCP sequence numbers.** These provide a sense of chronology to the TCP data sent and received. A telnet command or exchange might take several packets known as TCP segments to transmit all the data. Data is assigned a TCP sequence number to uniquely identify the data in each segment being sent. Because the data might arrive in a different order from which it was sent, TCP sequence numbers are also used to reassemble the data in the correct order.
- **Acknowledgements.** Acknowledgements are used to inform the sender that data has been received. Acknowledgements are made to sequence numbers to identify the exact data received. If the sender does not receive an acknowledgement for specific data in a given time, it assumes that the data has been lost. The sender will retransmit what it believes was lost.

2.5 ICMP

Internet Control Message Protocol (ICMP) was conceived as an innocuous method of reporting error conditions and issuing and responding to simple requests. Perhaps because of its seemingly benign origins, some of the current mutations of ICMP for less-than up standing purposes seem all the more outrageous.

In its pure state, ICMP is supposed to be a relatively simple and chaste protocol, but it has been altered to act as a conduit for evil purposes.

2.6 Rules

The rules in this project refers to the network intrusion signatures that we check in the data packets.

2.6.1 Headers

The rule header contains the information that defines the who, where, and what of a packet, as well as what to do in the event that a packet with all the attributes indicated in the rule should show up.

2.6.2 Protocols

The next field in a rule is the protocol. There are four protocols that we currently analyses for suspicious behavior:

- TCP, UDP, ICMP, and IP. In the future there may be more, such as ARP, IGRP, GRE, OSPF, RIP, IPX, etc.

2.6.3 Port Numbers

Port numbers may be specified in a number of ways, including any ports, static port definitions, ranges, and by negation. Any ports are a wildcard value, meaning literally any port. Static ports are indicated by a single port number, such as 111 for portmapper, 23 for telnet, or 80 for http, etc. Port ranges are indicated with the range operator :. The range operator may be applied in a number of ways to take on different meanings. Port negation is indicated by using the negation operator !.

2.6.4 **Options**

Rule options form the heart of intrusion detection engine, combining ease of use with power and flexibility. All rule options are separated from each other using the semicolon (;) character. Rule option keywords are separated from their arguments with a colon (:) character.

z	Domain Naming System
AS	Autonomous System
WEKA	Waikato Environment for Knowledge Analysis
ICMP	Internet Control Message Protocol
JVM	Java Virtual Machine
XP	Extreme Programming
JSF	Java Server Faces
JADE	Java Agent Development Framework
GUI	Graphical User Interface

LIST OF FIGURES

LIST OF TABLES

REFERENCES CITED

[1] A. K. Ghosh and A. Schwartzbard, a study of using neural network for anomaly and misuse detection, Proceedings of the 8th USENIX Security Symposium, page 12, Washington, D.C., USA, August, 1999.

[2] Brian C. Rudzonis. Intrusion Prevention: Does it Measure up to the Hype? SANS GSEC Practical vl.4b, 2003.

[3] DARPA. Knowledge Discovery in Databases, 1999. DARPA archive. Task Description http://www.kdd.ics.uci.edu/databases/kddcuip99/task.htm

[4] Axelsson S. The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection. In 6th ACM Conference on Computer and Communications Security, 1999.

[5] S. Bharadwaja, W. Sun, M. Niamat, F. Shen, Collabra: A Xen Hypervisor based Collaborative Intrusion Detection System, Eighth International Conference Information Technology: Next Generations, pp. 695-700, 2011.

[6] Johansen Krister and Lee Stephen. Network Security: Bayesian Network Intrusion Detection (BNIDS) May 3, 2003.

[7] Gregory F. Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. Machine Learning, 1992.

[8] Integrity in Automated Information Systems. National Computer Security, Center, September 1991.

[9] International Information Systems Security Certification Consortium, Inc. "(ISC) Code of Ethics." ISO/IEC 17799: Information Technology–Code of Practice for Information Security Management. 2000.

[10] Ramaswamy C., and Sandhu R., "Role Based Access Control Features in Commercial Database Management Systems," 21st National Information Systems Security Conference, Crystal City, Virginia, October 6-9, 1998.

[11] Ferraiolo D., Kuhn D., and Chandramouli R., "Role-Based Access Control," Artech House, Computer Security Series, 2003.

[12] The CISSP Prep Guide—Mastering the Ten Domains of Computer Security, Copyright © 2001 by Ronald L. Krutz and Russell Dean Vines. All rights reserved. Published by John Wiley & Sons, Inc.

[13] Jai Sundar Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, Eugene Spafford, Diego Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents", Purdue University.

[14] Wenke Lee, Salvatore J. Stolfo, "Data Mining Approaches for Intrusion Detection", Computer Science Department, Columbia University.

[15] Jake Ryan, Meng-Jang Lin, Risto Miikkulainen, "Intrusion Detection with Neural Networks", Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712

[15] Kruegel Christopher, Darren Mutz William, Robertson Fredrik Valeur. Bayesian Event Classification for Intrusion Detection Reliable Software Group University of California, Santa Barbara,, 2003.

[16] Brian C. Rudzonis. Intrusion Prevention: Does it Measure up to the Hype? SANS GSEC Practical vl.4b, 2003.

[17] DARPA. Knowledge Discovery in Databases, 1999. DARPA archive. Task Description http://www.kdd.ics.uci.edu/databases/kddcuip99/

[18] Kumar, Vipin, Jaideep Srivastava, and Aleksandar Lazarevic, "Managing cyber threats: Issues, approaches, and challenges".Vol. 5.Springer, 2006.

[19] Lei Li, De-Zhang Yang, Fang-Cheng Shen, "A Novel Rule-based Intrusion detection System Using Data Mining". In the Proc. Of 3rd IEEE International Conference on Computer Science and Information Technology, pp. 169-172, 2010.

[20] "Waikato environment for knowledge analysis (wek)." Available on: http://www.cs.waikato.ac.nz/ml/weka

[21] Data Mining Practical Machine Learning Tools and Techniques by Ian H Witten, Eibe Frank, Mark A Hall.

[22] Breiman, Leo, Friedman, J. H., Olshen, R. A., Stone, C. J., “Classification and regression trees”. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.ISBN 978-0-412-04841-8. (1984)

[23] G. Gu, P. Fogla, D. Dagon and W. Lee, “An Information-Theoretic Measure of Intrusion Detection Capability”. In Proceedings of the 2006 ACM Symposium on Information, computer and communications security; 21-24 Mar. (2006).

[24] Cannady J. (1998). Artificial neural networks for misuse detection. *Proceedings of the 1998 National Information Systems Security Conference (NISSC'98)*, 443-456, Arlington, VA.

[25] Lippmann, R.; Haines, J.; and Zissman, M. (2003). *An overview of issues in testing intrusion detection systems*. National institute of standards and technology (NTIS).

[26] Lia, L.B.; Chang, R.I.; Kouh, J.S. (2009). Detecting network intrusions using signal processing with query-based sampling filter. *Hindawi Publishing*

[27] Kang, B.D.; Lee, J.W; Kim, J.H.; Kwon, O.H.; Seong, C.Y.; Park, S.M.; and Kim, S.K. (2006). A mutated intrusion detection system using principal component analysis and time delay neural network. *LNCS*, 3973, 246 – 254.

[28] Grediaga, A.; Ibarra, F.; García, F.; Ledesma, B.; and Brotons, F. (2006). Application of neural networks in network control and information security. *LNCS*, 3973, 208–213.

[29] Rhodes, B.C.; Mahaffey J.A.; and Cannady, J.D. (2000). Multiple selforganizing maps for intrusion detection. *Proceedings of the 23rd National Information Systems Security Conference.*

[30] Mukkamala, S.; and Sung, A.H. (2003). Feature selection for intrusion detection using neural networks and support vector machines. *Transportation Research Record*, 1822, 33-39.

[31] Panda, M.; and Patra, M.R. (2007). Network intrusion detection using naïve bayes. *International Journal of Computer Science and Network Security (IJCSNS)*, 7(12), 258-263.

[32] 19.DARPA1998

[33] Horeis, T. (2003). Intrusion detection with neural network – Combination of selforganizing maps and redial basis function networks for human expert integration. http://ieee-cis.org/_files/EAC_Research_2003_Report_Horeis.pdf .

[34] Ramadas, M.; Ostermann, S.; and Tjaden, B. (2003). Detecting anomalous network traffic with self-organizing maps. *LNCS*, 2820, 36–54.

[35] Debar, H.; Becker, M.; and Siboni, D. (1992). A neural network component for an intrusion detection system. *IEEE Computer Society Symposium on Research in Security and Privacy*, 240-250.

[36] M. Chun Man, V. K. Wei, "A Taxonomy for Attacks on Mobile Agent", Proceedings of International Conference on Trends in Communications, Volume: 2, 2001, pp. 385-388.

[37] E. Royer, C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", IEEE Personal Communications, Volume: 6 Issue: 2, April 1999, pp. 46-55.

[38] J. Haines, L. Rossey, R. Lippmann, R. Cunningham,"Extending the DARPA Off-Line Intrusion Detection Evaluations", Proceedings of DARPA Information Survivability Conference & Exposition II, Volume: 1, 2001, pp. 35-45.

[39] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad-Hoc Networks", Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom'2000, pp. 275-283.

[40] LU Zhi-Jun, ZHENG Jing, HUANG Hao. A Distributed Real-Time Intrusion Detection System for High-Speed Network. Journal of Computer Research and Development, 2004, 41(4):667-673.

Sample Code

Home.java

```
import java.awt.Toolkit;
import java.io.FileInputStream;
import javax.swing.JFileChooser;
import javax.swing.UIManager;
import javax.swing.JOptionPane;
import javax.swing.JFrame;


public class Home extends javax.swing.JFrame   {

   public Home() {
      initComponents();
   }
   @SuppressWarnings("unchecked")
   private void initComponents() {

      HEADING = new javax.swing.JLabel();
      TRAIN_FILE = new javax.swing.JLabel();
      TEST_FILE = new javax.swing.JLabel();
      BTN_TRAIN_BROWSE = new javax.swing.JButton();
      BTN_TEST_BROWSE = new javax.swing.JButton();
      TXT_FLD_TRAIN = new javax.swing.JTextField();
      TXT_FLD_TEST = new javax.swing.JTextField();
      BTN_JUNCTION_TREE = new javax.swing.JButton();
```

```
jScrollPane3 = new javax.swing.JScrollPane();
jta2 = new javax.swing.JTextArea();
jScrollPane4 = new javax.swing.JScrollPane();
jta3 = new javax.swing.JTextArea();
jLabel1 = new javax.swing.JLabel();
BTN_NB = new javax.swing.JButton();
jScrollPane5 = new javax.swing.JScrollPane();
jtaNB = new javax.swing.JTextArea();
jLabel2 = new javax.swing.JLabel();
jScrollPane6 = new javax.swing.JScrollPane();
NB_RESULT = new javax.swing.JTextArea();
jScrollPane1 = new javax.swing.JScrollPane();
jTAMixed = new javax.swing.JTextArea();
jScrollPane2 = new javax.swing.JScrollPane();
jTAMixed1 = new javax.swing.JTextArea();
MixedBTN = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Framework for NIDS");
setBackground(new java.awt.Color(255, 255, 51));
setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
setForeground(new java.awt.Color(255, 51, 51));
setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("ids-eye.png")));
setResizable(false);

HEADING.setFont(new java.awt.Font("Times New Roman", 1, 24));
HEADING.setText("NETWORK INTRUSION DETECTION SYSTEM");
```

```
TRAIN_FILE.setText("TRAINING DATSET");


TEST_FILE.setText("TESTING DATASET");


BTN_TRAIN_BROWSE.setText("BROWSE");
BTN_TRAIN_BROWSE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BTN_TRAIN_BROWSEActionPerformed(evt);
    }
});


BTN_TEST_BROWSE.setText("BROWSE");
BTN_TEST_BROWSE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BTN_TEST_BROWSEActionPerformed(evt);
    }
});


BTN_JUNCTION_TREE.setText("Decision Tree");
BTN_JUNCTION_TREE.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BTN_JUNCTION_TREEActionPerformed(evt);
    }
});

jta2.setColumns(20);
jta2.setEditable(false);
jta2.setRows(5);
```

```
jScrollPane3.setViewportView(jta2);

jta3.setColumns(20);
jta3.setEditable(false);
jta3.setRows(5);
jScrollPane4.setViewportView(jta3);

jLabel1.setText("RESULTS SUMMARY");

BTN_NB.setText("NB");
BTN_NB.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BTN_NBActionPerformed(evt);
    }
});

jtaNB.setColumns(20);
jtaNB.setRows(5);
jScrollPane5.setViewportView(jtaNB);

jLabel2.setText("RESULTS SUMMARY");

NB_RESULT.setColumns(20);
NB_RESULT.setRows(5);
jScrollPane6.setViewportView(NB_RESULT);

jTAMixed.setColumns(20);
jTAMixed.setRows(5);
```

```
jScrollPane1.setViewportView(jTAMixed);


jTAMixed1.setColumns(20);

jTAMixed1.setRows(5);

jScrollPane2.setViewportView(jTAMixed1);


MixedBTN.setText("Mixed");

MixedBTN.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        MixedBTNActionPerformed(evt);

    }

});


jLabel3.setText("RESULTS SUMMARY");


javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup()

        .addGap(30, 30, 30)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

                .addComponent(TRAIN_FILE)

                .addGroup(layout.createSequentialGroup()

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .addComponent(BTN_JUNCTION_TREE)

                        .addComponent(jLabel1)
```

```
                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                                    .addComponent(jScrollPane4, javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(jScrollPane3, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 268, javax.swing.GroupLayout.PREFERRED_SIZE)))
                            .addGap(33, 33, 33)
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                .addComponent(jLabel2)
                                .addComponent(BTN_NB)
                                .addComponent(jScrollPane5, javax.swing.GroupLayout.DEFAULT_SIZE, 273, Short.MAX_VALUE)
                                .addComponent(jScrollPane6)))
                        .addGroup(layout.createSequentialGroup()
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                .addComponent(TEST_FILE)
                                .addComponent(TXT_FLD_TEST, javax.swing.GroupLayout.DEFAULT_SIZE, 421, Short.MAX_VALUE)
                                .addComponent(TXT_FLD_TRAIN, javax.swing.GroupLayout.PREFERRED_SIZE, 473, javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 26, Short.MAX_VALUE)
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                .addComponent(BTN_TEST_BROWSE)
                                .addComponent(BTN_TRAIN_BROWSE))))
                    .addComponent(HEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 528, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(33, 33, 33)
```

```
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
              .addComponent(jLabel3)
              .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 281,
Short.MAX_VALUE)
              .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 281,
Short.MAX_VALUE)))
          .addGroup(layout.createSequentialGroup()
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 253,
Short.MAX_VALUE)
            .addComponent(MixedBTN)))
        .addGap(34, 34, 34))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addComponent(HEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(TRAIN_FILE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(TXT_FLD_TRAIN, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(BTN_TRAIN_BROWSE))
            .addGap(14, 14, 14)
            .addComponent(TEST_FILE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(TXT_FLD_TEST, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(BTN_TEST_BROWSE))
            .addGap(31, 31, 31)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(layout.createSequentialGroup()
                .addComponent(BTN_JUNCTION_TREE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 221,
javax.swing.GroupLayout.PREFERRED_SIZE))
              .addGroup(layout.createSequentialGroup()
                .addComponent(BTN_NB)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jScrollPane5, javax.swing.GroupLayout.DEFAULT_SIZE, 226,
Short.MAX_VALUE))))
          .addGroup(layout.createSequentialGroup()
            .addComponent(MixedBTN)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 409,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel1)
          .addComponent(jLabel2)
          .addComponent(jLabel3))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 131,
Short.MAX_VALUE)
                    .addComponent(jScrollPane6, javax.swing.GroupLayout.DEFAULT_SIZE, 131,
Short.MAX_VALUE)
                    .addComponent(jScrollPane4, javax.swing.GroupLayout.DEFAULT_SIZE, 131,
Short.MAX_VALUE))
                .addGap(28, 28, 28))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void BTN_JUNCTION_TREEActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BTN_JUNCTION_TREEActionPerformed
        // TODO add your handling code here:
         try{
            JunTreeConn jc=new JunTreeConn();
            jc.init(trainpath, testpath, jta2, jta3);
            jc.run();
            BTN_JUNCTION_TREE.setEnabled(false);
        }
        catch(Exception e){
            e.printStackTrace();
        }
         BTN_JUNCTION_TREE.setEnabled(false);
    }//GEN-LAST:event_BTN_JUNCTION_TREEActionPerformed

    private void BTN_TRAIN_BROWSEActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BTN_TRAIN_BROWSEActionPerformed
```

```
        // TODO add your handling code here:
        trainpath = browse();
        TXT_FLD_TRAIN.setText(trainpath);


        BTN_TRAIN_BROWSE.setEnabled(false);
    }//GEN-LAST:event_BTN_TRAIN_BROWSEActionPerformed


    private void BTN_TEST_BROWSEActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BTN_TEST_BROWSEActionPerformed
        // TODO add your handling code here:
        testpath=browse();
        TXT_FLD_TEST.setText(testpath);
        BTN_TEST_BROWSE.setEnabled(false);
    }//GEN-LAST:event_BTN_TEST_BROWSEActionPerformed


    private void BTN_NBActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BTN_NBActionPerformed
        // TODO add your handling code here:
        try{
            NB nb=new NB();
            nb.init(trainpath, testpath, jtaNB, NB_RESULT);
            nb.run();
            BTN_NB.setEnabled(false);
        }
        catch(Exception e){
            e.printStackTrace();
        }
         BTN_NB.setEnabled(false);
```

```
  }//GEN-LAST:event_BTN_NBActionPerformed


  private void MixedBTNActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_MixedBTNActionPerformed
    // TODO add your handling code here:
    try{
      Compare cmp=new Compare();
      cmp.init(trainpath, testpath, jTAMixed, jTAMixed1 );
      cmp.run();
      MixedBTN.setEnabled(false);
    }
    catch(Exception e){
      e.printStackTrace();
    }
     MixedBTN.setEnabled(false);
  }//GEN-LAST:event_MixedBTNActionPerformed


  private String browse(){
    String path=new String();
    JFileChooser jfr = new JFileChooser();
        int check = jfr.showOpenDialog(this);
    if(check==JFileChooser.APPROVE_OPTION){
      path=jfr.getSelectedFile().getPath();
    }
    return path;
  }
  /**
  * @param args the command line arguments
```

```
*/
public static void main(String args[]) {
   java.awt.EventQueue.invokeLater(new Runnable() {
      public void run() {
         new Home().setVisible(true);
      }
   });
}
String testpath;
String trainpath;
NB nb=new NB();
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton BTN_JUNCTION_TREE;
private javax.swing.JButton BTN_NB;
private javax.swing.JButton BTN_TEST_BROWSE;
private javax.swing.JButton BTN_TRAIN_BROWSE;
private javax.swing.JLabel HEADING;
private javax.swing.JButton MixedBTN;
private javax.swing.JTextArea NB_RESULT;
private javax.swing.JLabel TEST_FILE;
private javax.swing.JLabel TRAIN_FILE;
private javax.swing.JTextField TXT_FLD_TEST;
private javax.swing.JTextField TXT_FLD_TRAIN;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
```

```
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JScrollPane jScrollPane4;
    private javax.swing.JScrollPane jScrollPane5;
    private javax.swing.JScrollPane jScrollPane6;
    private javax.swing.JTextArea jTAMixed;
    private javax.swing.JTextArea jTAMixed1;
    private javax.swing.JTextArea jta2;
    private javax.swing.JTextArea jta3;
    private javax.swing.JTextArea jtaNB;
    // End of variables declaration//GEN-END:variables

}
```