Development of a Network-based Intrusion Prevention System Using Data Mining Approach

تطوير نظام منع التطفل المعتمد على الشبكة بأستخدام

أسلوب تنقيب البيانات

By

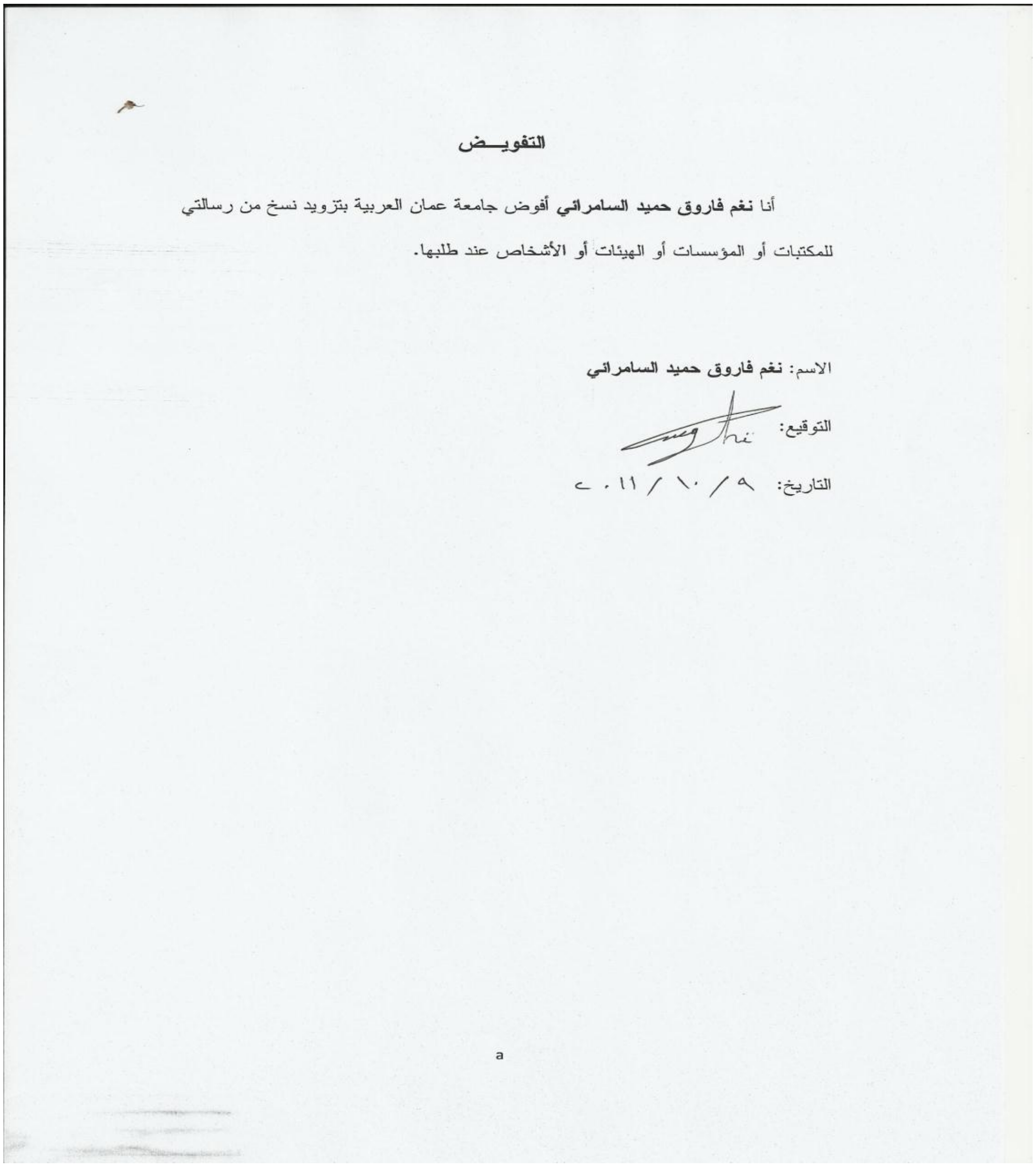
Nagham Farouk Al-Sammerai

Supervisor

Prof. Dr. Alaa Hussein Al-Hamami

A Thesis Submitted in Partial Fulfillment of the Requirements
For the Award of the Master's Degree in Computer Science

Department of Computer Science
College of Computer Sciences and Informatics
Amman Arab University
August 2011



التفويـــض

أنا **نغم فاروق حميد السامرائي** أفوض جامعة عمان العربية بتزويد نسخ من رسالتي للمكتبات أو المؤسسات أو الهيئات أو الأشخاص عند طلبها.

الاسم: **نغم فاروق حميد السامرائي**

التوقيع:

التاريخ: ٩ / ١٠ / ٢٠١١

جامعة عمان العربية
Amman Arab University

Resolution of the examining committee

This dissertation titled "Development of a Network-based Intrusion Prevention System Using Data Mining Approach" .Has been defended and approved on 17/9/2011.

Examining Committee	Title	Signature
Dr. Venus wizeer Simaoy	Chair	
Prof. Dr.Alaa Al-Hamami	Member and Supervisor	Alahamani
Dr. Malek Kakish	Member	

Dedication

I would like to dedicate this thesis to my parents who encouraged and supported me through all my study and the motivation for all I do.

Acknowledgment

All praise is due to Allah, for all the bless that he gave, without whose help, I would not have reached this far.

First and foremost, I am heartily thankful to my supervisor, Prof. Dr. Alaa Hussein Al-hamami, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. I am very lucky to have him as a supervisor as he is an experience lecturer.

I would like to express my gratitude to my beloved family who supported me. These are the people who have always given me a moral support in completing my study.

Lastly, I offer my regards and blessings to all of those who supported me and help me in any respect during the completion of the project.

Development of a Network-based Intrusion Prevention System Using Data Mining Approach

Abstract

Intrusion Prevention systems (IPS) can analyze, detect and prevent intruder attack. The IPS provides a good service in securing the network which is further the functionality than intrusion-detection systems (IDS), firewalls, antivirus and any security applications by actively responding to attacks and giving great flexibility when dealing with security threats.

The goal of improved NIPS based on both mechanisms is to detect patterns of known intrusions (misuse detection) and to distinguish anomalous network activity of intrusion from normal network traffic (anomaly detection) effectively. The Data mining methods have been used to enhance NIPS based on anomaly detection.

Using data mining methods lead to develop NIPS as an internal security gateway for defending against attacks and threats from inside and outside the computer network system. In addition, it will help to detect anomaly activity of suspicious probing inside the network before it launches any network attacks with damaging effects.

The study aims to enhance snort tool, which is NIPS base on both misuse and anomaly detection mechanisms, by using two sub-phases of data mining approaches, named improved K-mean clustering algorithm and PF-growth algorithm. The reason of a suggesting used these data mining approaches is due to the enormous volume of existing and newly appearing data that require

processing such as a snort log file, in addition it can help an analyst to discover new rules from a hidden patterns that snort tool cannot see as obvious rules.

Integration among these two sub-phases helps to discover new rules especially those related to internal network scans, besides unsupervised learning process in K-mean algorithm is used to discover new cluster may represent a new type of attack depending on decisions of analysts.

All that work, helps to enhance and to develop NIPS tool, by involving data mining approaches in investigating anomalies. Besides achieve objective to be a complete system performs requirements such as detect probe attack inside source of network and prevent it before launch network attack to the target machine with high performance, reduce false alarm, easy building system with low cast, and compatibility with any operating system. Furthermore, maximize the effectiveness in identifying attacks, thereby helping the users to construct more secure information systems.

تطوير نظام منع التطفل المعتمد على الشبكة بأستخدام

أسلوب تنقيب البيانات

Arabic summary

خلاصة

ان انظمة منع الاختراق لها قدرة على التحليل ،وكشف ومنع هجمات المتسلل .انظمة منع الاختراق يوفر خدمات جيدة في تأمين الشبكة التي هي اكثر وظيفيا من الخدمات التي تقدمها أنظمة كشف الاختراق،والجدران الحماية النارية،ومكافحة الفيروسات،وأي التطبيقات الأمنية من خلال الاستجابة بفعالية عند الهجمات واعطاء مرونة كبيرة عند التعامل مع التهديدات الأمنية.

الهدف من تحسين نظام منع التطفل الشبكة المعتمد على كلا من الآليات هو الكشف عن أنماط من الاختراقات المعروفة (باستخدام تقنية الكشف عن إساءة استخدام) والتمييز حركة الشبكة الشاذه للتسلل من حركة مرور الشبكة العادية (باستخدام تقنية كشف عن الشواذ) وعلى نحو فعال. وقد استخدمت أساليب تنقيب البيانات من أجل تحسين نظام منع التطفل الشبكة المعتمد على تقنية كشف الشواذ.

باستخدام أساليب تنقيب البيانات يؤدي إلى تطوير نظام منع التطفل الشبكة باعتبارها بوابة الأمن الداخلي للدفاع ضد الهجمات والتهديدات من داخل وخارج نظام شبكة الكمبيوتر. بالإضافة إلى ذلك ، سوف تساعد على الكشف عن النشاط المشبوه الشاذ للاستطلاع الذي يحدث داخل الشبكة قبل أن تطلق أي هجمات الشبكة مع تأثيرات مدمرة وخطيرة.

هذه الدراسة تهدف إلى تحسين برنامج snort ، وهو نظام منع التطفل الشبكة المعتمد على كلا من الآليات كشف سوء الاستخدام وآليات الكشف عن الشذوذ ، وذلك باستخدام اثنين من نهج تنقيب البيانات ، وهما خوارزمية المحسنة من K-mean وخوارزميةFP-growth . وسبب اقتراح استخدام هذه الأساليب من تنقيب البيانات ويرجع ذلك إلى الحجم الهائل من البيانات الموجودة والبيانات الحديثة والتي تتطلب معالجة مستمرة مثل ملف السجل الخاص ببرنامج snort ، بالإضافة لها يمكن أن تساعد المحلل لاكتشاف قواعد جديدة من نمط الخفية التي برنامج snort يغفل عنها ولا يمكن أن يرى فيه قواعد واضحة.

توحيد ودمج بين هذه العميلين يساعد على اكتشاف قواعد جديدة خاصة تلك المتعلقة باستطلاع شبكة الاتصال الداخلية ، بالإضافة إلى منهج عملية التعلم غير خاضعة للرقابة المستخدمه في خوارزمية K-mean تساعد في اكتشاف كتلة جديدة قد تكون هي ممكن ان تمثل نوعا جديدا من الهجوم يعتمد على قرار من المحللين.

كل هذا العمل ، ويساعد على تعزيز وتطوير برنامج نظام منع التطفل الشبكة ، التي تنطوي على نهج استخراج البيانات في التحقيق في الحالات الشاذة. بالاضافة الى تحقيق الهدف المراد تلبية متطلبات نظام منع التطفل الشبكة مثل كشف هجوم التحقيق التي مصدرها من داخل الشبكة ومنعها قبل شن هجوم الشبكة إلى الجهاز المستهدف مع الأداء العالي ، والحد من انذار كاذب ، وسهلة التنفيذ مع انخفاض الكلفة ، والتوافق مع أي نظام تشغيل. علاوة على ذلك ، أقصى قدر من الفعالية في تحديد الهجمات ، مما يساعد المستخدمين على بناء نظم معلومات أكثر أمنا.

Table of Contents

List of Abbreviation

ADAM	Audit Data Analysis and Mining
API	Application Programming Interface
DBA	Distribution Based Artificial Anomaly
DDoS	Distributed Denial of Service
DLL	Dynamic Linked Libraries
DoS	Denial of Service
ER	Evidential Reasoning
FCMs	Fuzzy Cognitive Maps
FP	Frequent Pattern
FTP	File Transfer Protocol
HIPS	Host Intrusion Prevention System
IBL	Instance Based Learning
IBM	International Business Machines
ICMP	Internet Control Message Protocol
IDDM	Intrusion Detection using Data Mining
IDP	Intrusion Detection and Prevention
IDS	Intrusion Detection System
IIDS	Intelligent Intrusion Detection System
IOWA-ADCPRID	IOWA-Automated Discovery of Concise Predictive Rules for Intrusion Detection
IP	Internet Protocol
IPS	Intrusion Prevention System
ISS	Internet Security System
JAM	Java Agents for Meta learning

LINQ	Language Integrated Query
LVQ	Learning Vector Quantization
MADAM ID	Mining Audit Data for Automated Models for Intrusion Detection
MCDA	Multi-Criteria Decision Analysis
MINDS	Minnesota Intrusion Detection System
Netsh	Network shell
NIPS	Network Intrusion Prevention System
NNID	Neural Network Intrusion Detector
OS	Operating System
PAM	Partitioning Around Medias
R2L	Remote to Local user
RFCs	Request For Comments
RIDS	Rising Intrusion Detection System
RTT	Round-Trip Time
SOM	Self Organizing Maps
SQL	Structured Query Language
TCP	Transmission Control Protocol
TFN	Tribal Flood Network
U2R	User to Root
UDP	User Data gram Protocol
VB	Visual Basic
WinPcap	Windows Packet Capture
XML	Extensible Markup Language

List of figures

CHAPTER ONE
INTRODUCTION

1.1 Introduction

The most important points and challenges facing the technology are the information security and preservation of its integrity from any intrusion. In spite of the huge development in technology and networks with increased value of the information stored accompanied by an increase in software and hardware products that specializes in security and trying to maintain the integrity of this information but was accompanied by an increase in the risk of intrusion and hacker exploited weaknesses in the security products.

Recently, most of the information about an individual is stored online by companies and government organizations; for example, a finance company and mortgage can keep information on customer financial credit rating, social security number, bank account numbers, and a lot more personal information of the customer.

The Intruders may break into the system and copy data, and the user never knows. Therefore, the damage from digital personal data loss may be far greater than loss of physical data; also, damages caused by a hacker either breaking into a network or using a computer to launch an attack on another networks are possible [1].

Computer attacks become more sophisticated and skilled; organizations today are keenly aware of the need to provide effective security and protect their information system, and existence of networks requires the protection of the gateway and the nodes.

Although many security applications are available such as Intrusion Detection Systems (IDS), Intrusion Prevention System (IPS), anti-virus/Spam systems; firewalls have been proposed to control the attacks, securing distributed systems and networks is still extremely challenging.

Each one of these security applications covered one component of the total network security picture. Some of these are distinguished by features such as detection without prevention, the use of a mechanism either the anomaly-based or signature-based detections/preventions, the ability to defense the network from outside threats, but it is a limitation to detect threats coming from inside the network computer system. While other tools focused to improve latency *"is the time it takes to respond and take appropriate action, this period of time is critical in the success of an attack"* without taking into account an undesirable increase false negative *"is any malicious traffic that makes it through the security applications to the production network"* or false positive *"is any legitimate traffic that the security applications drops because it appears to be anomalous"* [2].

IPS can analyze, detect and prevent intruder attack, which further the functionality of IDSs, firewalls, antivirus and any security applications by actively responding to attacks and giving great flexibility when dealing with security threats.

IPSs are typically used on the outer boundary of a network to prevent any malicious traffic from reaching possibly vulnerable systems inside the network that may contain sensitive information.

Currently many research tended to improve the work of IPS for computer network, some focused on the types of mechanism detection and prevention, and the other focused on choosing the best architecture of IPS for network.

This research will use one of IPS system, which use rule (signature)-based prevention Integrated with data mining, which cans detect and expect anomaly attack .As well as the capacity and effectiveness of data mining in dealing with the huge information stored in the database of the network and high speed processing search and extract the appropriate decision.

Network administrators can determine the security policy violations using analysis of enough data collected. Unfortunately, the data is so huge even for a small network and traditional methods of analysis so time consuming and difficult even with computer assistance because foreign features can make it harder to detect suspicious behavior patterns, complex relationships exist between the features, which are practically impossible for humans to discover, the solution to this problem is using data mining approach [3, 4].Using Intrusion Detection/prevention and data mining are capable of working together efficiently to provide network security.

The objective of this proposed system will be the internal gateway of the network protects the computer network as well as all its hosts from signature and anomaly attacks, able to detect all threats, especially those that occur inside the network. Furthermore, it will decrease the load of the intrusion prevention task on every individual host, reduce the possibility human errors that occur during contribution to make the right decision and try to achieve the desired reduction of the rate of false positives, false negatives and minimized latency.

1.2 Intrusion Prevention Systems (IPSs):

Intrusion Prevention Systems (IPSs) any device (hardware or software) that has the ability to monitors, detect attacks and activities for malicious or unwanted behavior and can react in real-time to prevent the attack from being successful [2].

1.2.1 The importance of an Intrusion Prevention System

Companies know that information is an extremely valuable asset, but many people fail to suitability protect that information from unauthorized people. While technology has given us many beneficial possibilities, it has also put sensitive information at risk.

Any Enterprises without security strategy prevent its information from external and internal threats open the door to unacceptable risks, costs, undesired access, malicious content, and rate-based attacks.

The main objective of security techniques system reviews network weaknesses and takes steps to maintain security and protect information corporate assets and intellectual property from spyware and other intruders

Many techniques are strong to a defense in depth approach to security such as firewalls and anti-virus programs try to block attacks and IDS tries to identify attacks as it occurs, but have limitations.

A firewall can stop suspicion services or attacks by blocking certain port numbers but it does little to analyze and to evaluate traffic that uses allowed port numbers. IDS can evaluate traffic that passes through these open ports but cannot stop it or prevent attacks. With the proliferation of sophisticated attacks and the discovery of new vulnerabilities, new processes are needed to protect precious data and network resources [5, 6].

As well as mostly used of security techniques based on signature approaches that focus on how an attack works, trying to detect certain strings. If the attacker makes minor changes, then stored written signatures no longer detects the attack.

As a result previously security techniques as IDS do not prevent attacks either network or host type, they just silently monitor the traffic and trigger an alarm when attack is detected without stopping or even slowing down progress of attack [6].

Intrusion Prevention Systems, next generation of security systems, use new proactive process not just to detect attacks, it tries to stop intrusions before any damage is done ,distinguish unauthorized activity from normal activity and has a set of signatures or predefined conditions that, when match, effect a response [5].

IPSs are proactive approaches that have the ability to drop packets or even disconnect connections before accessing to the host and block all traffic with the same IP source, when IPSs detect illegal activity; they rapidly stop the intrusion and minimize the overall time before the network is back to normal by using multiple detection methods.

The benefit of IPS position in the line of network traffic is that is can detect attacks and intrusions more accurately and reliably through less dependence on signatures and more on intelligent methods of detection, so the IPS generates far fewer false alarms [6].

The summary of actions taken by IPS when observes any suspicion activity:

- Generating alarm.
- Preventing attack activity.
- Resetting the connection.
- Modifying firewall rules.
- Dropping attack packet and allowing pass other normal packet.
- Logging the event activity and updating the log event database.

1.2.2 The requirements of successful IPSs

There are some requirements commonly used when evaluating the fine tuning of IPS or any security techniques that can be further used to analyze the successfulness of IPS.

- Accuracy: the most important requirement in an IPS is accuracy. Having false positives must be absolutely unacceptable in an IPS. *A false positive is any legitimate traffic that will drop because it appears to be anomalous*. False positives are commonly generated by security systems that depend on a single detection method, and by ones that

- cannot be configured at different levels to fit into the operational environment. If legitimate traffic is blocked, then problems appear for authorized users. This creates DoS (Denial of Service) attacks that originate from the prevention system itself. For example a valid business transaction may act like an attack. In such a case, this packet may first be dropped and then the entire data flow and may be the source is critical business and the recipient will be prevented from accessing resources.

- Performance: the importance of IPSs is performance. One of the problems with IPS is that it tends to occur a network bottleneck. Network traffic needs to flow through IPS to be analyzed and if they don't operate quickly enough, they drop packets or pass packets, increasing the possibility of false negatives. *A false negative is any malicious traffic that makes it pass through the IPS to the production network*. Thus, IPSs have to work equated with line of speed.
- Flexibility: Ability of prediction Unknown Attacks and Easy Signature Update for New Attacks. An IPS system must provide flexible methods to update new attack signatures constantly, as well as these systems should have capabilities to deal with entirely new classes of attacks without depending on database signature updates. IPSs use methods such as inverse exclusion method where all given destination requests except that legal are dropped. Protocol validation method, where illegal protocol request are dropped. Another method is attack-independent blocking where hostile attackers are identified, and all traffic from the attacker is dropped, regardless of whether the attacks are known or not.

- Reliability and Availability: An IPS system should be reliable and high available. Reliability refers to the ability of a system to perform its functions properly without interfering with other systems on the network also IPSs should cooperate with these systems such as firewalls, antivirus systems, etc. While the availability is the amount of downtime of the system, due to shutdown, crashes, or maintenance. An IPS gives the network security administrator many facilities; it is capable of detecting attacks and intrusions and directly affects limiting or blocking network traffic. IPSs have an easy interface for setting and changing configurations on its system.

- Minimize Latency: *Latency is the time it takes for a packet to pass through the IPS to the destination system and return to the user*. This is typically measured in Round-Trip Time (RTT). With all the necessary time to analyze and detect the content of packet before being sent to the destination system.

1.3. Intrusion detection /prevention methods

Two main approaches have been devised to detect intruders. Misuse Detection depends on the signatures of some known attacks, whereas Anomaly Detection depends on other attacks and only reflects some deviation from normal patterns.

1.3.1Misuse Detection

This type of detection is based on the knowledge of system vulnerabilities and known attack patterns. Misuse detection is concerned with finding intruders who are attempting to break into a system by exploiting some known vulnerability.

An intrusion detection system continually compares recent activity to known intrusion scenarios to ensure that one or more attackers are not successful to exploit known vulnerabilities. There are many techniques of misuse detection but difference between these techniques is in how they describe or model the behavior that constitutes an intrusion.

The misuse detection systems used rules to describe events refer to intrusive actions that a security administrator looked for within the system. Large numbers of rules may be difficult to interpret within detection system. Misuse detection system use rule organizational techniques including model-based rule organization and state transition diagrams to overcome difficulties of large numbers of rules by use the rules to look for events that possibly fit an intrusion scenario. The events may be monitored live by monitoring system calls or later using audit records [7].

1.3.2.Misuse detection systems

Misuse detection based also known as knowledge based, signature based, pattern matching. This system activates an alarm when a match is found to a signature contained in known signature database of attacks. These attack signatures in the audit data are based on a set of rules that match typical patterns of exploits used by attackers.

- Simple systems or pattern match. This is the simpler way to detect Misuse attack signatures (known attacks) scan for specific byte sequences (signatures) that stored in known attacks database but
- requires frequent update of its database to avoid false positives [8].

- Expert systems or stateful matching scans for attack in context of a traffic stream or code misuse signatures as if-then implication rules rather than scans individual packets (byte sequences). Signature analysis focuses on defining specific descriptions and instances of attack-type behavior to flag and detects signatures spread across multiple packets. But also requires frequent update of its database to avoid false positives [8].

- Data mining techniques can be used to detect consistent and useful patterns of system features that describe program and user behavior, and use the set of related system features to compute (inductively learned) classifiers that can recognize known intrusions.

- Model based reasoning attempts to combine models of misuse signature with Evidential Reasoning (ER) "*is a generic evidence-based Multi-Criteria Decision Analysis (MCDA) approach for dealing with problems having both quantitative and qualitative criteria under various uncertainties including ignorance and randomness*" to support conclusions about the occurrence of a misuse signature. This technique may be useful for identifying intrusions which has different audit paths patterns.

- Keystroke monitoring is a simple hardware or software technique that monitors the action of tracking the keys struck on a keyboard for identifying attack patterns.

So the misuse detection system is more accurate because there is a database of known attacks so lead to avoid high rate of false positives. But this system suffers some problems as:

- The system cannot detect unknown attacks that have no matched patterns or unachieved signature stored in its database. Insider attacks may also go undetected.
- The system be responding after a new attack has occurred because cannot predict new attacks.
- The database of this system has to be updated continuously to keep up with new type of attacks.

1.3.3. Anomaly Detection

This type of detection assumes that an intrusion will always reflect some deviations from normal patterns; anomaly detection can be divided into static and dynamic anomaly detection.

A static anomaly detector is based on the assumption that there is a portion of the system being monitored that does not change. The static portion of a system is the code for the system and the constant portion of data upon which the correct functioning of the system depends. Static anomaly detectors focus on integrity checking.

While Dynamic anomaly detection typically operate on audit records or on monitored networked traffic data. Audit records of operating systems record events of interest. Therefore only behavior that results in an event that is recorded in the audit will be observed and these events may occur in a sequence.

In distributed systems, partial ordering of events is sufficient for detection. In this case; thresholds are defined to separate normal resource consumption from anomalous resource consumption [7].

1.3.4. Anomaly detection systems

These systems compare observed activity against expected normal usage profiles (for users, groups of users, applications, etc). These profiles define a baseline for normal user tasks. Audit event records which fall outside the definition of normal behavior are considered anomalies.

1. Threshold monitoring sets metric values for defining acceptable behavior. Thresholds provide a clear, understandable definition of unacceptable behavior, but it is difficult to establish suitable threshold values and time intervals over which to check and its result in a high rate of false positives or false negatives.

2. Statistical anomaly deals with profiles that characterize baselines of normal system or user traffic activity and throughput, and triggers an alarm for deviations from those baselines. While Traffic anomaly seeks to watch for unusual traffic activities, such as a flood of UDP packets or a new service appearing on the network, both approaches can identify unknown attacks and DoS floods and must have a clear understanding of normal traffic environment to tune properly [8].

3. User work profiling maintains individual work profiles to which the activities user is expected to do in the future rather than little deviation from the expected normal such as the time being longer than usual usage, recent changes in user work patterns and irregular user requests. While group profiling that covers group of users with a common work pattern, resources requests, usage these resources, and historic activities. Group profiling is expected that each individual user in the group follows the group activity patterns.

4. Executable Profiling deals with monitor executable programs use the system resources, especially those whose strange deviations of activity are traced to a particular originating user. Viruses, Trojan horses, worms, trapdoors, logic bombs and other such software attacks are addressed by profiling how system objects such as files, printers and any resource are normally used by users, and also by other system subjects on the part of users.

5. Rule-based approach is creating rules by analyzing normal traffic is a complicated task. These rules represent normal user behavior or profiling (user, group, resources, etc) .Another approach is Protocol anomaly detection which falls under this category and analyzes packet flows, looks for deviations from standards set forth in RFCs. Reduces false positives with well-understood protocols but may result in high rate of false positives and false negatives with poorly understood or complex protocols [8].
6. Neural networks approaches are trained by presenting them with a large amount of data, and rules about data relationships to determine if traffic is normal or not. And can be used to improve the performance of intrusion detection in anomaly detection with a high detection rate and a low false positive rate.

The advantage of anomaly detection system is no need for signatures database to detect previously unknown attacks and insider attack. The limitations of this system are:

- The system will classify any anomalous activities as intrusion even activities not intrusive to lead to high rate of false positives alarm generated which are due to legitimate activity.
- This type of system is also computationally expensive because the overhead of keeping track of requires a lot of work to understand, building, and updating several system profile metrics.

1.4.Types of Intrusion Prevention Systems

There are two types of Intrusion prevention system that use one of the intrusion detection methods; IPS fall into two categories:

- Host-based systems base their decisions depend on information obtained from a single host. Software product used for this type.
- Network-based systems base their decision by obtaining data monitoring the traffic network to which the hosts are connected. Such systems are typically a hardware product.

1.4.1Network-Based Intrusion Prevention Systems (NIPSs)

Network Intrusion Prevention System (NIPS) are known as monitoring device and are considered as an Intrusion Detection System (IDS) inline with the firewall without suffering from latency (the time it takes for the IDS to either modify the firewall rules or issue a TCP reset command in model bundling up IDS and firewalls, this period of time is critical in the success of an attack).

NIPS are used as a great way to prevent attacks from happening on the network. The NIPS checks every packet that passes through it, analyzing traffic for known attack patterns designed to infect, disable, or take over another computer system. When a pattern is matched and the NIPS detect an attack, it takes an action as alert, log, send reset, typically modifying firewall rules or blocking the corresponding packet stream, prevent the attack from happening again, generating a notification that lead to prevention successful intrusions [1, 2].

Usually, NIPSs are inline and sit between the network traffic flow between two or more network interfaces and monitor network traffic at a collection point; they make a response to an event happen almost immediately. The true power of NIPS is in their capability to dynamically block the offending traffic [8].

NIPS architecture varies from product to product, but there is a basic underlying structure to all. These include system service scanner, traffic normalize, detection engine, and traffic shaper [1].

- System service scanner uses to build a reference table that provides the knowledge of the target system.

- Traffic normalization is the first process facing traffic network to intercept and resolve the traffic that has abnormalities before it sends to evaluated for malicious code in the next stage. This process involves discard the packet that does not conform to the set security policy criteria or patterns based on protocol states and blocking traffic based on the criteria that would be put in a firewall [1]. The normalization also may hold packet fragments and reassemble them into a packet based on its knowledge of the target system using System service scanner because if these fragmented packets slip through the network traffic and are reassembled at the host, will appear as abnormal traffic and will disable it. Traffic Normalization is

- the process of removing exploitable ambiguities and ensures that traffic interpreted by the NIPS is the same as that seen by the end host [5].

- Traffic after leaving the previous stage enters into the detection engine handles all pattern matching that is not handled by the normalization.

- The last stage before the traffic leaves the NIPS enters into traffic shaper for classification based on traffic protocol and flow management or in the future will be based on user and applications.

A. Network-Based Intrusion Prevention Systems Benefits

1. Zero Latency Prevention. As a hardware device the NIPSs reduce this high latency by providing the notification within one circuit instead of two.

2. Effective Network Hygiene. Since many attacks whose signatures are known, NIPS remove these packets quickly.

3. Simplified Management. This is due all packaged into one hardware; it reduces storage space and overall management.

B. Network-Based Intrusion Prevention Systems limitations

1. It does not do much effective against anomaly Attacks.
2. High Availability. May not be able to withstand high traffic availability and tolerance needed by all first and head-on network devices.

3. Detection effectiveness. It has not yet been tested for effectiveness of detection and it does not stop everything.

4. Production Readiness. This occurs because the technology is new and has not gotten the field-testing it needs to prove effectiveness in every test.

1.4.2. Host-Based Intrusion Prevention Systems (HIPSs)

HIPS, is Intrusion prevention system designed for security over host-based where intrusions and infections are dealt with at the individual workstation level to provide a more effective level of security.

A HIPS usually sits between the kernel and utility software application that sends requests to the kernel of the Operating system. Actions of the HIPS include blocking the request which an intrusion activities and denying access to the kernel [5].

HIPS based on one host, is work by simulation software which a process of restricting the definition of acceptable behavior rules used on HIPSs. HIPS prevention occurs when simulation software intercepts system calls or system messages by utilizing dynamic linked libraries (dll) substitution. The substitution is accomplished by injecting existing system dlls with vendor stub dlls that perform the interception. So when calls made to system dlls actually perform a jump to vendor stub code where then the bad calls are processed, evaluated, and dealt with. Most the simulation software at the kernel level of the system because processes system calls can be intercepted easily [1].

A. Host-Based Intrusion Prevention Systems Benefits

- Effective simulation context-Based prevention. HIPS use simulation context to protect the host, they have complete context of the host environment, and are therefore more capable of dealing with such attacks.
- Effective against anomaly Attacks. Since HIPS can define acceptable parameters application or operating system service behavior to enable the agent to prevent any malicious attack on the host.

B. Host-Based Intrusion Prevention Systems limitations

- Deployment Challenge. There are difficulties in deploying the remote agents on every host. These hosts need updating.
- Difficulty of Effective simulation software Configuration. It can be a challenge to define effective and nonrestrictive parameters on hosts.

Lack of Effective Prevention. HIPS cannot prevention signature attacks.

1.4.3. IPSs Approaches

There are some of IPS methods being used that presented as the following:

- Protocol anomaly detection is used to ensure that packets meet to the protocol requirements and have no ambiguities. Protocols should be well defined, this lead to high accuracy detection of the deviations from the protocol standard. For example, by IP spoofing of FTP PORT commands, the attacker can tell the FTP server to open a connection to a victim's IP address and then transfer a Trojan horse to the victim. Checking for a match between the IP address in the FTP PORT

- command and the client’s IP address can prevent this anomaly.
- Traffic anomaly detection is operating on the basis of deviations from expected behavior. Attackers often use a port or network scan as a precursor to an attack and the scanning techniques that used by attackers have made it possible that worms can affect the entire vulnerable of system in 10s of seconds or less, so fast that no traditional Anti-attack response is possible. NIPS implement throughput and threshold triggers that alert to such scanning activity, increasing the possibility that prevented an attack.

- State-based signature detection is based on the context specifie by the user, looks at related portions of traffic by tracking state, to detect attacks. It is not completely automated as the user needs to have previous knowledge about the attack. For example the Love Letter worm can be detected by a rule that would read as follows: “Look for ‘ILOVEYOU’ in the subject field only, ignore this string anywhere else in the email”. Of course false positives can be generated in this case, since harmless emails with the same title may have been sent.

- Pattern matching using regular expressions use to detect attack patterns that are slightly different from the fixed ones because the simply change like a space or a tab in the attack code could be enough to avoid detection. So regular expressions provide wild-card and complex pattern matching, and are able to prevent attacks.

- Signature detection is used in cooperation with the above mentioned techniques to prevent combined attack types seen on today's networks.

- Hybrid approach Typically used in NIPS , is use various detection methods, including protocol anomaly, traffic anomaly, and signature detection work together to determine an imminent attack and block traffic coming from an inline router.

- Software-based heuristics: This approach usually using on HIPS is similar to anomaly detection system using neural networks to act against new or unknown types of intrusion.

- Sandbox approach: use on HIPS is a Mobile code like ActiveX, Java applets or any scripting language is quarantined in *a sandbox, an area with restricted access to the rest of the system.* This system then runs the suspect mobile code in the sandbox and monitors its behavior. If the code not meets a predefined security policy, it is stopped and prevented from executing.

- Kernel-based protection: Typically used in HIPS. Kernel based IPS prevent execution of malicious system calls. The kernel controls access to system resources like memory, input/output devices and CPU. Programming code errors enable exploits as buffer-overflow attacks to overwrite kernel memory space and crash or take over computer systems. To prevent these types of attacks, a software agent is loaded between the user application and the kernel.

- The software agent intercepts system calls to the kernel, inspects them against an access control list defined by a policy and then either allows or denies access to resources.

- Address space randomization is a technique used to fortify systems against buffer overflow attacks. The idea is to introduce artificial diversity by randomizing the memory location of certain system components, and checking whether the code about to be executed by the operating system came from a normal application or an overflowed buffer, these attacks can be stopped.
- Protecting System Resources – used to prevent alteration of system resources by hacking tools such as Trojan horses, root kits, and backdoors and can change in system resources like libraries, files/directories, registry settings, and user accounts. This system disallows install hacking tools.

- Stopping Privilege Escalation Exploits – Privilege escalation attacks try to take ordinary users root or administrator privileges. This type can prevent change privilege levels, disallowing attacks access to resources, and block exploits.

- Prohibit Access To E-mail Contact List – Many worms spread by mailing a copy to those in the Outlook‘s contact list. This approach could be preventing these worms by prohibiting e-mail attachments from accessing Outlook’s contact list.

- Prevent directory traversal – An approach that would prevent the hacker access to the web server files outside its normal range could prevent malicious activities. Sometimes the directory traversal has vulnerability in different web servers that allow the hacker to access files outside the web servers range.

1.5.The Importance of using Data Mining for Intrusion Detection/ Prevention Systems

Data mining is the process of examining data to uncover patterns and deviations as well as determining any changes or events that have taken place within the data structure [3]. Its can improve a network intrusion detection system by adding a new level of observation to detection of network data in differences and identifying the boundaries for usual network activity so it can distinguish common activities from uncommon activities [9].

Data mining improves intrusion detection/prevention system using a variety of different methods [9]:

1. Code Variants: data mining is based on the process of scanning for abnormal activity through code variants instead of unique signatures. For example, a buffer overflow whose code has been changed would be considered a fraud by attempting to escape an intrusion detection system that uses signatures.

2. Data Reduction: Data mining can significantly reduce data overload through its capability to extract specific amounts of data for identification and analysis [10]. This helps the system to determine which data is most relevant and breaks it down so anomaly detection is easier and faster in execution time and processing.

3. Filter out Valid Network Activity: Data mining is used to help intrusion detection by being able to better identify valid network activity so it can filter it out to make detection of abnormal activity in data easier.

4. Attacks without Signatures: Since data mining is not signature-based like intrusion detection, it is more efficient in detecting abnormalities that do not contain signatures. If network activity contains a specific profile and rules of protocol, an abnormality is easily detected and can be extended to individual hosts, entire networks, specific users, and overall traffic patterns on the network at specific times.

1.6.Data Mining and Intrusion Detection /prevention system

Data Mining is powerful assisting for most applications that required data analysis. Recently, data mining is becoming an important component in intrusion detection /prevention system.

Data mining could contribute to the enhancement of the applications of network intrusion Detection/prevention systems, data mining use one or more techniques are used in the context of intrusion detection which analyze network data to gain intrusion related knowledge, such as:

1. Data summarization with statistics, including finding outliers that lead to finding anomalous activity that discovers a real attack.

2. Clustering, including segmentation of the data into natural categories that lead to identifying different IP address has same activity, this ongoing pattern can be a type of attack.

3. Association of rule discovery, including defining normal activity and enabling the discovery of anomalies that help to separate normal activity from suspicion data to allow focus on real attacks

4. Classification, including predicting the category to which a particular record belongs data, this identify which data generate alarm and attack signatures.

Usually using data mining techniques to analysis of collected data in an offline database, this important in performing Network Intrusion Prevention Systems (NIPS) because all connections have already finished therefore these techniques can process and check all features without drop packets when flooded with data became faster than process, as well as offline database provides the ability to transfer data from multiple hosts to central host for analysis ,detection and prevention that a way to increase the performance and the accuracy of Network Intrusion Prevention Systems (NIPS) [3,5].

1.6.1. Clustering

Human labeling of network audit data instances or even used traditional methods are time-consuming and expensive because these amounts of available data is huge therefore begins used clustering approach which is a technique for statistical data analysis used in many fields such as machine learning and data mining [3].

Clustering is the process of labeling data and assigning it into groups. Clustering algorithms can be partition the data set into subsets or clusters; so that the objects in each cluster share some common feature often proximity according to some defined distance measure [11].

Clustering techniques can be categorized into:

- Pairwise clustering, pairwise clustering unifies similar data instances based on a data-pair wise distance measure.
- Central clustering classes, while central clustering that also called centroid-based or model-based clustering, models each cluster by its “centroid”, and more efficient than Pairwise clustering algorithms.

Clustering is used to detect attack in any cluster that modeled according to pre-defined metrics and common features of sets data belonging to this cluster by discovering complex intrusions occurred over extended periods of time and different spaces, correlating independent network events and in anther mean the Clustering is useful in intrusion detection as attack activity should cluster together, separating it from normal activity [11, 12].

One of the common clustering techniques is K-means clustering which used to find natural groupings of similar alarm records; this depends on the records that are far from any of these clusters indicate unusual activity that may be part of a new attack.

Most of the clustering techniques are the basic steps involved in identifying intrusion. These steps are as follows:

1. Find the largest cluster, i.e., the one with the most number of instances, and label it normal.
2. Sort the remaining clusters in an ascending order of their distances to the largest cluster.
3. Select the first K1 "no. of clusters" so that the number of data instances in these clusters sum up to ¼ ´N, and label them as normal, where ´ N is the percentage of normal instances.
4. Label all the other clusters as attacks.

1.6.2Classification

Classification is similar to clustering in that it also partitions data records into distinct segments called classes. But it differs from clustering, classification require more because it need also labeling data set for training

stage and classification is much less exploratory than clustering because the end-user decides on the attribute to use define the classes and each record has a value for these attribute, so classification is not used to explore the data or approximate its values to discover interesting segments but to assign new data has a specific value to pre-defined categories or classes [3,12].

A Classification based IDS even IPS attempts to classify all traffic as either normal or abnormal class, this technique has been popular to detect individual attacks, but has been suffered the problem of high false positives and false negatives rate so begin applied with complementary fine-tuning techniques to reduce its troubles [11,12].

Classifications algorithms can be classified into three types:

- Extensions to linear discrimination (e.g., multilayer perceptron, logistic discrimination).
- Decision tree and rule-based methods (e.g., C4.5, AQ, CART).
- Density estimators (Naïve Bayes, multi-Bayes, k-nearest neighbor, LVQ (Learning Vector Quantization), SOM (Self Organizing Maps)).

Data classification for intrusion detection can be achieved by the following basic steps.

1. First to learn the classification models of the normal and abnormal system call sequences, it needs to supply it with a set of training data containing pre-labeled normal and abnormal sequences. The mechanism models based on any type of classification algorithms, all these can be used to scan the normal network paths and create a list of unique sequences of system calls. This list is generally named as normal list.
2. Next the second step is to scan each of the intrusion paths. For each sequence of system calls, first look it up in the normal list. If an exact match can be found then the sequence is labeled as normal, otherwise it is labeled as abnormal.

3. Finally must ensure that the normal paths include nearly all possible normal short sequences of system calls, because an Intrusion path contains many normal sequences in addition to the abnormal sequences since the illegal activities only occur in some places within a network path.

Classification technique in the domain of intrusion detection or prevention system needs the large amount of data needed to be collected to apply classification. To build the traces and form the normal and abnormal groups, significant amount of data need to be analyzed to ensure its proximity. Using the collected data as empirical models, false alarm rate in such case is significantly lower when compared to clustering.

In intrusion detection, data mining classification can be applied to a standard set of malicious virus and benign executable using derived features, then classification approach can be useful for both misuse detection and anomaly detection, but it is more commonly used for misuse detection.

Thus the various classification approaches can be employed on network data for obtaining specific information and detecting intrusion and then prevention, for example the Naïve Bayes and multi-Bayes classifiers can be used to detect malicious virus code.

While the decision Tree can be exploited to formulate genetic algorithm to create rules that match a set of anomalous connection.

Nearest neighbor classifier approaches based on SOM and LVQ can be used to refine the collected network data in intrusion detection [12].

1.6.3Association Rule

The Association rule is particularly designed using in data analyses. Association rule mining finds interesting associations or correlation relationships among huge set of data items. Association rule shows attribute value conditions that occur frequently together in a given dataset [3, 12, and 13].

The association rule considers each pair (attribute/value) as an item. In each single network request an item set is a combination of items .The algorithm scans through the dataset trying to find item sets that tend to appear in many network data. The objective behind using association rule based data mining is to derive multi-feature (attribute) correlations from a database table.

Association rules construct information or rules in the form of "if-then" statements. Association rules are probabilistic in nature. In addition to the antecedent (the "if" part) and the consequent (the "then" part), an association rule has two numbers that express the degree of uncertainty about the rule. In association analysis the antecedent and consequent are sets of items that are disjoint. The first number is called the support for the rule. The support is simply the number of transactions that include all items in the antecedent and consequent parts of the rule. The other number is known as the confidence of the rule. Confidence is the ratio of the number of transactions that include all items in the consequent as well as the antecedent to the number of transactions that include all items in the antecedent [12].

Many association rule algorithms can be classified into two categories:

- Candidate-generation-and-test approach such as Apriori.
- Pattern-growth approach.

Association rule algorithms are multiple scans of transaction databases and a large number of candidates therefore became use association rule in analyzing network data in intrusion detection [14].

Basic steps for integrating association rule for intrusion detection as follows:

- First network data have to be constructed into a database table where each row is an attribute record and each column is a value field of the attribute records.
- There is index that intrusions and user activities shows frequent correlations among network data. For example, “program policies”, which codify the access rights of privileged programs, are concise
- and capable to detect known attacks is in that the intended behavior of a program, e.g., read and write files from certain directories with specific permissions is very consistent. These consistent behaviors can be captured in association rules.

- With the association rule, rules based on network data can continuously merge the rules from a new run to the aggregate rule set of all previous runs, and then can get the capability to capture behavior in association rule for correctly detecting intrusion and hence lowering the false alarm rate.

1.6.4Outlier Detection

An outlier is an uncommon observation that significantly deviates from the characteristic distribution of other observations. The value of outliers indicates that individuals or groups that have very different behavior from most of the individuals of the dataset. Many times, outliers are removed to improve accuracy of the estimators. Outlier detection has many applications, such as data cleaning, fraud detection and network intrusion.

Anomaly detection algorithms require a set of purely normal data to train the model [15, 16]. Assume that anomalies can be treated as previously unobserved patterns. Since an outlier may be defined as a data point which is very different from the rest of the data, than can employ several outlier detection schemes for intrusion detection which are based on statistical measures, clustering methods and data mining methods.

Commonly used outlier techniques in intrusion detection are Mahalanobis distance, detection of outliers using Partitioning Around Medias (PAM), and Bay’s algorithm for distance-based outliers. Outlier detection is very useful in anomaly based intrusion detection. With outlier detection approach, can detect novel attack/intrusion by identifying them as deviation from normal behavior [17, 18].

The basic steps in detecting intrusion based on outlier detection are as follows [12]:

1. As outlier detection technique is used in anomaly detection, first step have to identify normal behavior. This behavior can be data set or pattern of some events on the network.

2. Then useful set of feature need to be constructed.
3. And similarity function needs to be defined between them.

4. Also will need to run specific outlier detection algorithm on the set of feature. The algorithm can be based on a statistical based approach, a distance based approach, or a model based schema. All these approaches are based on finding the deviation between collected and scanned data sets.

5. In case of intrusion detection, the collected set of data set will be the set of events and their relation to intrusion. Such relation can be calculated based on normal behavior and any other behavior which significantly deviates from normal behavior. As with such deviation we can preempt attacks based on their behavioral deviation. Outlier detection approaches can useful for detecting any unknown attacks.

This is the primary reason that makes outlier detection a popular approach for intrusion detection systems. Statistical based outlier detection scheme uses a probabilistic model as representation of underlying mechanism of data generation. Such probabilistic model can be useful in intrusion detection environment to decide the probability before alarming the system for intrusion.

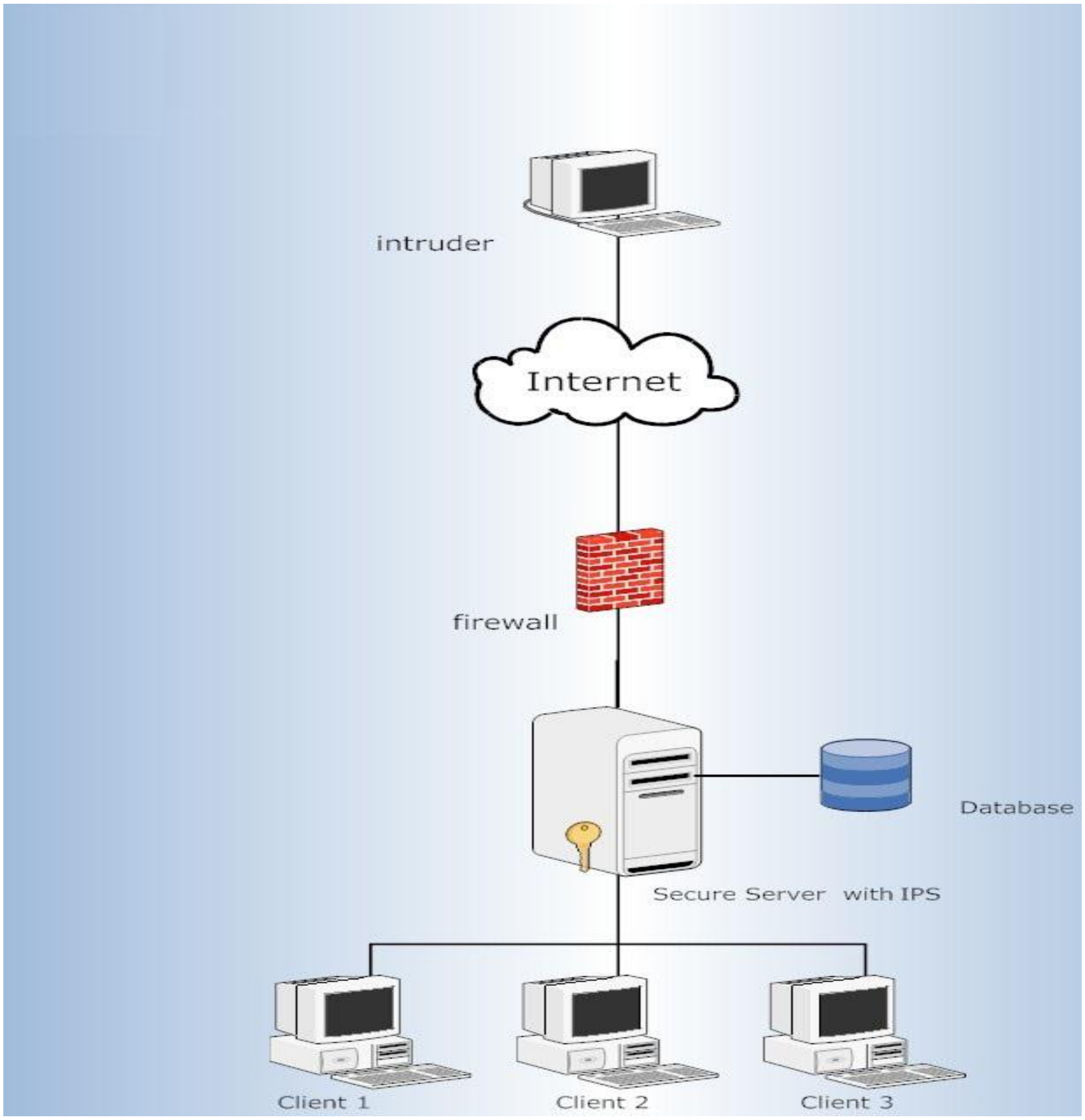
So the outlier detection is very useful in anomaly based intrusion detection systems that are involved in detecting abnormal behavior or deviating patterns. Its can help to identify abnormal behavior from the set of normal behavior and enable to detect any unknown intrusions [19].

1.7 The Problem statement

The future security for Network Intrusion Prevention Systems is became the need for a system has mechanisms be able to detect all types of attack, what is known signature or not. A system need to increase protection for the network from inside by enhancement the ability of detection the internal threats as same efficiency protected the network from external threats. Furthermore, a system be able analyze and extract information from huge data and identify what is really anomaly behavior of attack effectively with reduce false positive or negatives alarms. In addition to that it minimized latency by increasing the ability of detection. This leads to a high-speed response and prevents intrusions before access to any target in the system.

The aim of this study is to present a system consisting of Network Intrusion Prevention System (NIPS) which uses a signature approach that drops malicious traffic based on a set of pre-specified rules and also use anomaly approach but with weakness in detection a new of an anomaly attack . It will be integrated with data mining which has the ability to detect, anticipate anomaly attack effectively and extract specific amounts of data from huge database for identification and analysis which lead to faster in execution time and processing compared with search in large data and delays in responding to the prevention of intrusion. The proposed system will be gateway to the network and protect computer network as well as all of its hosts from any internal or external threats as in Figure 1-1

Figure 1-1: general network diagram with proposed system of NIPS.



1.8 Thesis Contribution

This system will contribute to improvement significantly in the following:

- Cooperation between the data mining and IPS helps to reduce the effort on the IPS base on network when high traffic availability.

- Decrease the load of the intrusion detection/prevention tasks on every individual host.
- Using sever database to store all profile behavior signature and new anomaly attacks when detect which help to solve difficulties in deploying the mobile agents on every host and also solve problem update in each host.
- Maximize the level of security network as well as all of its hosts from known and unknown attacks.
- Using data mining methods achieving reduce the rate of false positives, false negatives and minimized latency.
- Using data mining techniques to analysis an event log file in an offline database, this is important in performing Network Intrusion Prevention Systems (NIPS) because all connections have already finished. Therefore, these techniques can process and check all features without drop packets when flooded with data became faster than a process. As well as an offline database provides the ability to transfer data from multiple hosts to central host for analysis, detection and prevention that a way to increase the performance and the accuracy of Network Intrusion Prevention Systems.
- Make system can analyze traffic and determine whether normal traffic or not, even if not normal traffic the system can determine is attack traffic or just traffic has special features cause by user and reduce human assistance in make the decision.

1.9. Thesis organization

The thesis consists of five chapters; each one handles a certain subject as the following:

Chapter one presents an introduction about the study, It deals with the basic concept in the study which is: security, a brief about IPs, the importance of an IPS in terms of network security, the requirements for designing successful IPS, the methods of IPS, the benefits and limitations for each type of IPS, motivated the need to use data mining for IPS, the overview of data mining techniques that involved in identifying an intrusion. This chapter gives the overview about expected improvement in case a success of the proposed system.

Chapter two will talk about the history of Intrusion Prevention Systems, understanding the concept of IPS through understanding the network attack types in detail. And also it will talk about the history of used Data mining approach as a tool in IDS/IPS. It will discuss related work of IPS tools that combine misuse and anomaly intrusion-detection techniques by using data mining and the limitations of these tools will be taken into consideration to improvement the proposed system that being performed.

Chapter three will explain the important protect the network from internal attacks and dangerous for reconnaissance or probing inside network. And it will show the proposed system design, and described in details the tool used in building up the system and role that it plays in the system.

Chapter four shows the methodology of implementing the proposed system, starting from sniffer network traffic and analyzing it through the NIPS tool's rules that takes the right decision in pass or drops this traffic and records the events in a log file, passing through processing the log file of NIPS using data mining algorithms to discover the hidden behavior of network traffic between the source and destination to find what the NIPS tool is overlooked it, and generated candidate rules to build improvement NIPS actions. In addition this chapter will explain the testing scenarios and display the output results.

Chapter five will discuss the conclusion and future works.

CHAPTER TWO

OVERVIEW AND PREVIOUS WORKS

2.1 Introduction

Revolution of security technology is not new or recent discovery but it adds to the development of the attack types that become intelligent, sophisticated, more accurate, and more identification in its target. Therefore, this led to be a widening gap between traditional security applications that used and new attacks. Given the increase of the Internet activity, larger numbers of computers are being corrupted every day by attacks, became growing interest in complementary network-level security mechanisms in an attempt to reduce this gap, as provided by firewalls, network intrusion detection, prevention systems and development of these applications.

Now, there is no standalone security program or mechanism that can guarantee completely a secure network. The Network administrators use a variety of networks, and host based tools, including firewall, Intrusion Detection Systems (IDSs), patch and version managers, and anti-virus tools to keep an acceptable level of security and to deal with the constant attacks in the network environment. These tools combine together to form an integrated line of defense against network attacks.

Many types of intrusion exist, which are various in methods of action and constantly evolving to be more intelligent against detected by security tools. The increased availability of broadband network means that computer viruses, worm and any type of attack can spread at a rate faster than ever before, as well as the increase in the range and volume of attacks spreading via the internet lead to protection against attacks is becoming more difficult. Even confidential data is stored on servers detected by fraud that performed from remote location. The successful access to the data of system network by attackers can result in loss of confidentiality, integrity and availability of data, system and services [20].

There are three approaches to network security [20, 21], which are:

- Protection through filtering known unwanted traffic.
- Protection through assessment by using tools testing contentious for unknown vulnerabilities.
- Protection through detection that implementing measures to detect unwanted traffic a network can be secured.

Today, the most commonly used security strategy is to use end host based solutions that rely on security tools, such as antivirus software, firewall, HIDS, HIPS and so on. The main problem of these approaches is the inability to protect thousands of hosts in less than an hour [20]. Using security tools based network as solution such as firewall, NIDS and NIPS to protect the server and all hosts of these servers in network.

Firewall usually inspects the packet headers to determine whether the packets allowed to pass through or dropped. However, firewalls are not effective to protect networks from worms, viruses and intrusion.

The NIDS need to examine both the headers and the payloads of each incoming packet for thousands of suspicious patterns to identify attack signatures. That is differs NIDS from a firewall.

The NIDS is able to discover attempts intrude by hackers that use malicious attacks but is the inability to prevent damage network happened by attacks.

NIPS is capable of detecting possible malicious packets within normal traffic, taking a predefined action to stop intrusions and to prevent illegal traffic from passing on network before it does any damage, and alerting on that intrusion while action does or after it, the malicious packet has been delivered and deal with it. While NIDS can detect and alert on a possible intrusion without take any prevent action. The actions of NIPS are key different from NIDS [20]

.Unlike firewalls or other security applications, network intrusion prevention systems (NIPSs) are significantly more performance, accuracy, activity in detection and prevention attack, complex and, as a result, are use as software or hardware device with routers and firewalls in the network-level security.

The complexity produces from the need to analyze not just headers but also packet content and higher-level protocols. Moreover, the function of NIPSs needs to be updated with new detection components, due to the continuously development nature of network attacks, usually NIPS suffer from false alarm. The complexity, accuracy and performance are address challenges of NIPS.

Data mining has significant advantage in work in huge data, activity analyze and determine the required pattern effectively, so a lot of research and companies are beginning to move towards the use of data mining technique in improving the performance of network intrusion prevention systems.

NIPSs are proactive defense mechanisms to be able to prevent malicious activity the system must first be able to detect such activity. NIPS is must be more effective and can protect a large area of network with one device. NIPS must be done in real time environment to provide the best possible measure of intrusion prevention [20].

With realization now network security must be defense in depth, tends to system has begun to depending on concept that never relies on a single defensive technique. So NIPS must be enhancement to cover all defensive mechanism of security network.

2.2 History Intrusion Prevention Systems (IPSs)

First start with a brief historical overview and the most important feature of each period:

- 1970s : use Rudimentary audit-trail analysis.
- 1980s : use Rules-Based expert systems.
- 1990s : Burst revolution of research IDS systems.
- 2000s
  - Appearance of Active IDS
    - Intrusion Detection and Prevention (IDP).
    - Intrusion Prevention Systems (IPS).

- Combination of Technologies in one package
    - Firewall + IDP + Anti-Virus.
    - Devices and Security Switches.

Currently, the hacker's tools to launch network attacks are easily on hand. Furthermore, information about security vulnerabilities and tools uses to trade and exchange among Hackers. This means that even the person doesn't have to be a security expert to write hacks and code tools to launch an attack, he can simply to use these tools are readily available.

In addition, the professional hackers have increased in their capabilities greatly over the last few years. Figure 2-1 illustrates the rise in hacking expertise over the years

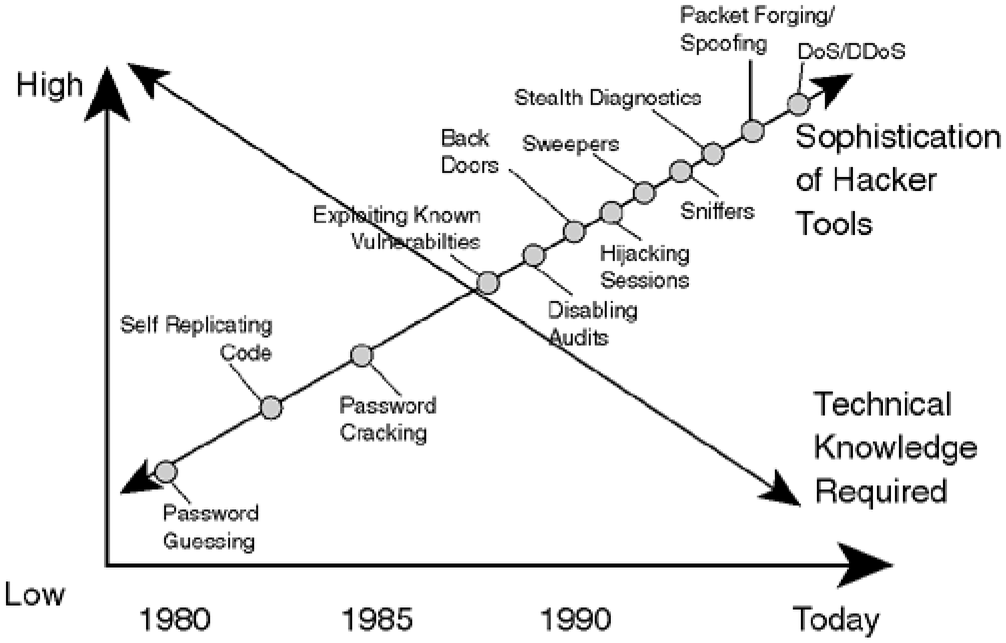


Figure 2-1 Rise in Hacker Capabilities

The first IPS's were invented independently by Jed Haile and Vern Paxon [23] in the early 2000s to resolve what was believed to be ambiguities in passive network monitoring done by firewalls and IDS's. By placing detection systems in-line and IPS could make access control decisions based on application content, rather than access control list filtering based on IP address or ports as traditional firewalls had done. This step is considered as improvement upon firewall technologies.

The term intrusion prevention system was originally created by Andrew Plato, a technical writer and consultant for Network ICE, now part of IBM's Internet Security Systems group, and the original creators of the first commercially available IPS, Black ICE. "Intrusion prevention" technology is considered by some to be an extension of Intrusion Detection (ID) technology, but it is actually another form of access control, like an application layer firewall [23].

Black ICE came on to the market in 1998. Both a business and personal version of the product were offered. Black ICE was able to provide both host-based and network-based IPS capabilities using protocol analysis. The Black ICE products included a firewall that could respond, in real-time to intrusions and block attackers. Network ICE was purchased in June 2000 by Internet Security Systems (ISS). ISS purchased by IBM in 2006

2.3 The network attack type

In order to understand the concept of how intrusion detection can work to prevent some of these attacks, it is important to look at some common cases of such attacks. Understanding these attacks helps understand the overall concept of IPS [19].

A network intrusion is often called a network attack. It can be categorized into two main types of attacks based on mode of attack:

- Denial of Service (DoS)
- Network access attacks can be further subdivided into two categories:
    - Data access.
    - System access.

One important type of activity that often precedes most network access and DoS attacks is reconnaissance or probing which is in some research classified as type of attack.

1. Probing:

An attacker surveys a target network or system by sending various types of packets and finds the vulnerabilities in a network's security posture .It begun 1970s – early 1980s .This can be done by using automated tools (such as SATAN, network mappers, port mappers, Ipsweep) that carry out port sweeps across networks, trying to find machines that might be susceptible to a particular type of attack. This process can also be carried out manually by an experienced attacker.

The general process of reconnaissance usually involves a series of steps, starting with a hacker scouting a network to find out the network's general parameters, such as the perimeter devices, IP addresses, traffic patterns, and domain names. The next step involves trying to verify which ports on various machines might be available to launch various types of attacks. Other information of the various machines can also be found at this point, such as the topology of a target network, the types of network traffic allowed through a firewall, the active hosts on the network, the operating systems are running in those hosts, the server software they are running and the software version numbers for all detected software. Unfortunately, from the perspective of someone performing a scan, they are legally scouring the internet to find publicly accessible resources.

Probing does not penetrate but its providing information help to penetrate, by divulging critical information regarding the machine's vulnerabilities that the hacker can exploit.

The IDS and IPS are usually able to differentiate between legitimate and malicious scanning. Scanning is the most common attack as it is the precursor to any serious penetration attempt.

2. Denial of Service Attacks

Generally, a DoS attack is an attacker disables services that a service provider offers its users .The Appearance of a DoS attack is back to the late eighties. Most DoS attacks can be classified into two categories:

a. Resource exhaustion

It attacks attempts to stop the system's normal functioning by consuming all its resources that are available on a network system. This makes the system unable to process an order of legitimate users because it has run out of resources by attack. Most of these attacks target a network's CPU resources or the bandwidth in terms of connection speeds. This type of attack can be divided into a large number of categories. However, generally classify as simple DoS attacks and distributed DoS attacks.

- Simple DoS attacks are launched by a single attacker against one or more victims such as TCP SYN floods, smurf attacks, and various packet storms.
- Distributed Denial of Service (DDoS) attacks are organized against victim machines by a large number of attacking machines that are compromised and forced to help launch various types of DoS attacks such as the Feb 7-11, 2000 attacks and SubSeven attack.

b. Attacks Designed to Cause Immediate Cessation of Normal OS Operations

It is an attack in which exploited vulnerability in the OS or a protocol in order to stop the OS's functioning abruptly and completely through the malicious content of a packet sent to the system. Examples of these types of attacks are ping of death attacks, UDP bomb, and land.c attacks.

3. Network access attacks

In network access attacks, an intruder gains unauthorized access to resources on a network and uses this access to carry out any number of unauthorized or even illegal activities .this type is back to the late eighties. Generally classify as Data access attacks and System access attacks.

Data access: is known as privilege escalation. Unauthorized data retrieval involves reading, writing, copying, or moving files that are not intended to be accessible to the intruder. An example is when someone gains access to system files on a Web server where the aim was to give that person access to only the files in the published area of the Web server.

System access: is a more annoyed form of network access attack in which an attacker gains access to system resources and devices. This access can include running programs on the system and using its resources to do things as commanded by the attacker. Such as brute-force password attacks, Trojan horse attacks, and various attacks using tools to exploit weaknesses in the software code running on a machine.

An intruder can assess to resources on a network in two ways:

- R2L: unauthorized access from a remote machine and usually used with guessing password attack.

- U2R: unauthorized access to local super user (root) privileges and often used with various ``buffer overflow'' attacks.

Examples of Attacks

This section looks at the several examples of various attacks to get a better understanding, how they work and how Can IPS detect and lead to prevent these attacks.

1. Ipsweep

Ipsweep is an observation sweep to determine which hosts are listening on a network. This information is useful to an attacker in launching attacks and searching for non-immune machines.

There are many methods that can be used to perform an Ipsweep .The most common method and the method used within the simulation is to send ICMP Ping packets to every possible address within a subnet and wait to see which machines respond. The Ipsweep probes in the simulation were not stealthy the sweeps were performed linearly, quickly and from a single source.

Prevention Ipsweep an IPS determines the Ipsweep used by looking for many Ping packets, destined for every possible machine on a network, all coming from the same source.

2. TCP SYN Floods

A TCP SYN flood is a simple DoS attack. It is also known as a half-open SYN attack. It is sending a large number of TCP SYN packets to a server faster than the system can process them. These packets have a source IP address that is spoofed and not in use. When the server receives these requests, it responds to them using SYN-ACK. However, because the source IP address is spoofed and unused, the TCP handshake is never completed. (It is completed only if the starting machine sends an ACK to the server upon receiving the SYN-ACK.) .The server waits for the ACK. For this, it must allocate resources and buffer space to record the information it has received in the SYN packet and sent out in the SYN-ACK packet in response

. This causes the memory to fill up, forcing the new connections to be ignored. This leads the memory to fill up, forcing the new connections to be ignored as described in Figure 2.2.

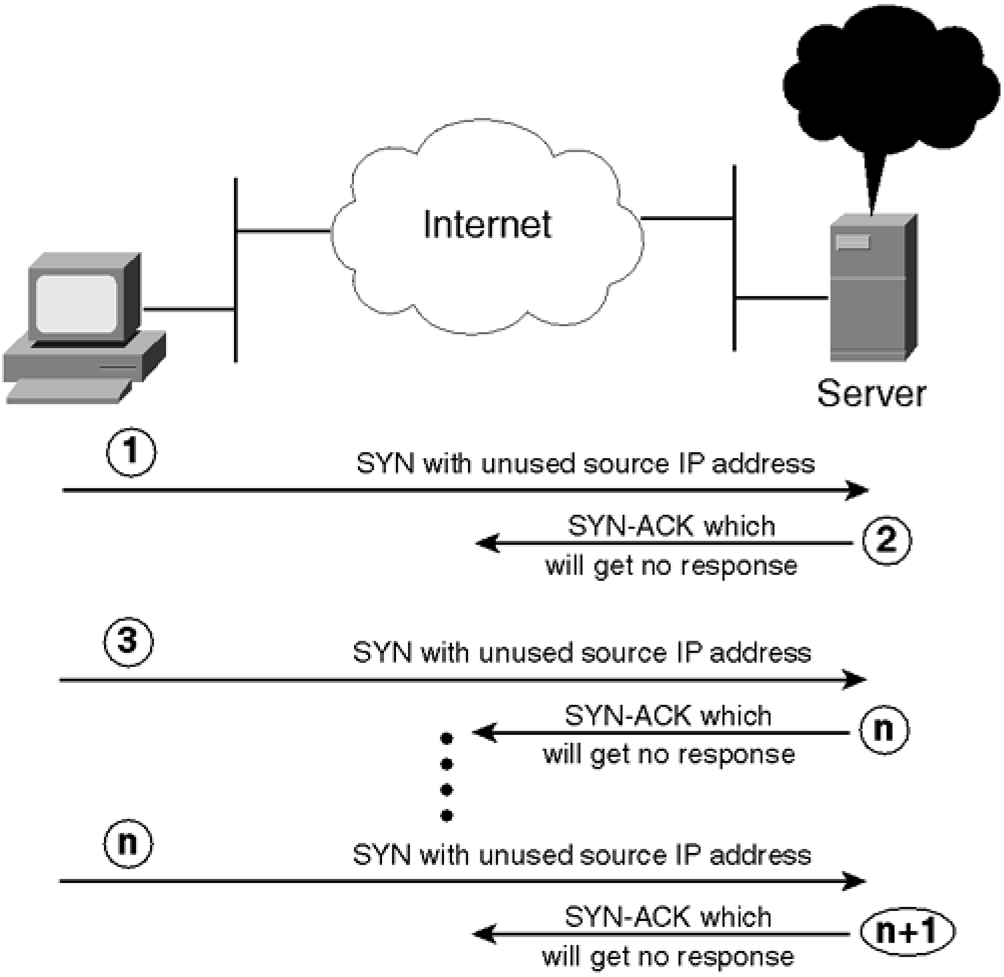


Figure 2-2 TCP SYN Flood DoS Attack

To prevent this attack, use the detection triggers whenever a large number of SYN packets are seen in a short period of time. However, There is a product of a false positive alarm when its trigger incorrectly. For example, if a busy website becomes unavailable for a few minutes, then is brought back online, this event triggers because of the "pent up" connections waiting for the system to become available.

Most network-based IDSs can detect SYN floods and reset these connections, freeing up resources on the servers. IDSs can achieve this by looking for patterns of activity giving away SYN flooding.

3. The February 7-11, 2000 Attacks

These DDoS attacks were a combination of four types of DDoS attacks (Trinoo, TFN (Tribal Flood Network), TFN2K, Stacheldraht), which are brought down some major commercial Web sites on the Internet down for extended periods of time during a week.

These four types of DDoS attacks based on a network of master/slave programs that coordinate with each other to launch DoS attack against a victim machine. Although the general structure is same in each one, but the details of how these tools are implemented communication between the masters and slaves can easily be modified by a hacker. For example, it is simple to modify the TCP ports.

The TFN attack is more complicated than the UDP flooding that Trinoo uses. These attacks include ICMP flooding, SYN flooding, and smurf attacks. In TFN2K, attacks are launched using spoofed IP addresses, making detecting the source of the attacks more difficult than in TFN attacks.

Stacheldraht allows communication between the attacker and the masters (called *handlers*) to be encrypted. In addition, slave (called *agents*) can upgrade their code automatically. The Stacheldraht agents can launch ICMP flood attacks, SYN floods, and UDP floods. The attacks can be conducted with or without spoofing. The agents send test packets with spoofed addresses to see if the network edge routers for the network on which they are installed allow spoofed addresses. If they don't, they use only last-octet spoofing. The architecture of the Stacheldraht network shows in Figure 2-3.

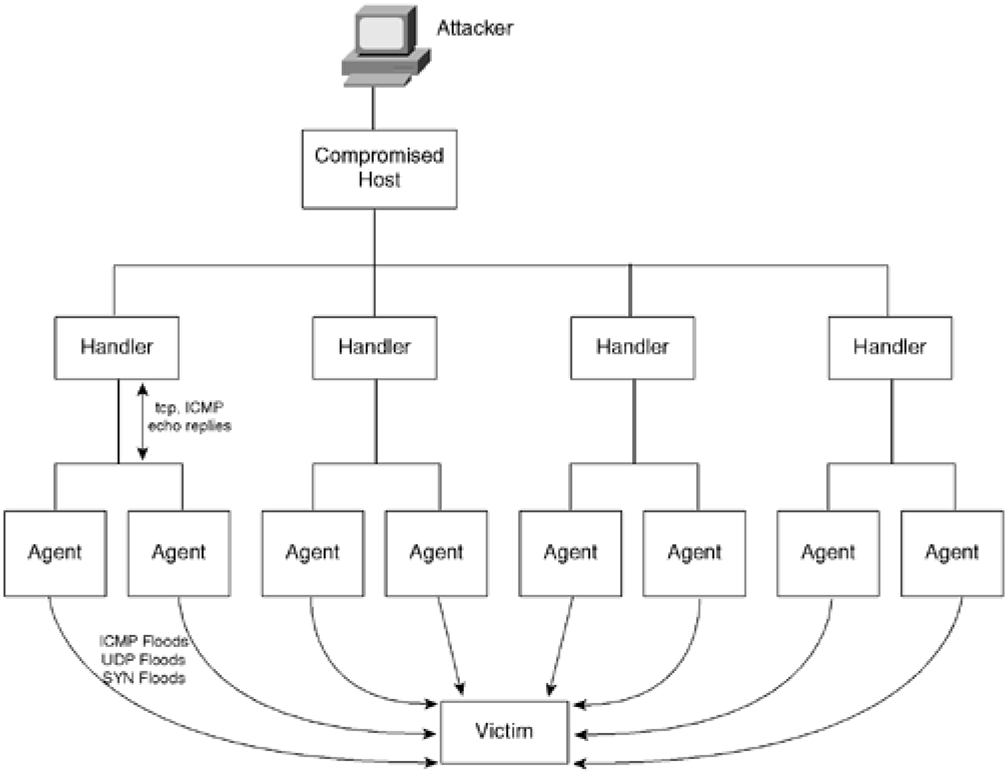


Figure 2-3. Stacheldraht Attack

NIDS can be used to detect the four types of tools just described after they have been set up on a network. This detection is based primarily on the telltale fingerprints of communication between the masters and slaves. Because all these tools can use echo replies (except for Trinoo) to communicate, the signatures watch for echo replies for which they do not see an echo going out first.

For example IDS signatures used to detect Trinoo networks by looks for UDP packets containing potential commands from a Trinoo client to a server, and looks for UDP packets containing potential command replies from a Trinoo server to a client.

4. Ping of Death Attack

A ping of death is a denial of service attack that attempts to crash your

system by sending a fragmented IP packet. IP does not allow single packets to exceed 65536 bytes, but the fragments themselves can add up to more than that.

A hacker can send an IP packet to a vulnerable machine such that the last fragment contains an offset where (IP offset * 8) + (IP data length) > 65535. This means that when the packet is reassembled, its total length is larger than the legal limit, causing the buffer overruns in the machine's OS. This attack is generally carried out by sending an ICMP packet encapsulated in an IP packet. Thus, it is called a ping of death attack. This leads the operating systems crash when they receive this data.

NIDSs can generally recognize such attacks and drop it by looking for packet fragments that have the IP header's protocol field set to 1 (ICMP), the last fragment bit set, and (IP offset * 8) + (IP data length) > 65535. This implies a packet in which the IP offset (which represents the starting position of this fragment in the original packet, and which is in 8-byte units) plus the rest of the packet is greater than the maximum size of an IP packet.

5. Land.c Attack

It is a DoS attack in which an attacker sends a host a TCP SYN packet with the source and destination IP address set to the host's IP address. The source and destination port numbers are the same as well. Upon receiving a SYN packet, the host responds with a SYN-ACK to itself. Because this is theoretically impossible, Windows goes into an infinite loop trying to resolve these illegal connections, causing the whole network performance to be degraded.

Network-based IDS implementations can detect land attacks and drop such the malicious packets once identified, by looking for IP packets that have a source IP address equal to the destination address. This is generally called an impossible IP packet.

6. Buffer Overflows

A buffer overflow attack is orchestrated by sending to an OS data that is too large for the relevant buffer handling the data to store. Buffer overflow attacks can be classified into the categories of both DoS and network access attacks. The reason for this is that although buffer overflows can cause operating systems on network devices to crash and cause a denial of service, they can also be exploited by attackers to gain access to the operating system and go further than simply causing a DoS attack.

A simple form of buffer overflows attack is also called *smashing the stack*. As it is known that an OS's buffer is co-located with other critical parts of the memory; including the memory location that contains the pointer to the next memory location that the OS is to go to after the program using the buffer is done. Therefore, important pieces of information get overwritten.

An attacker can send a large piece of data that is constructed so that the memory location containing the pointer to the next memory area to go to get overwritten by a value that refers to a memory location that contains code that the attacker wants to execute. This code can allow the hacker to get more privileges on the computer, or the access can simply be used to crash the system completely. The buffer overflow attack is illustrated in Figure 2-4.

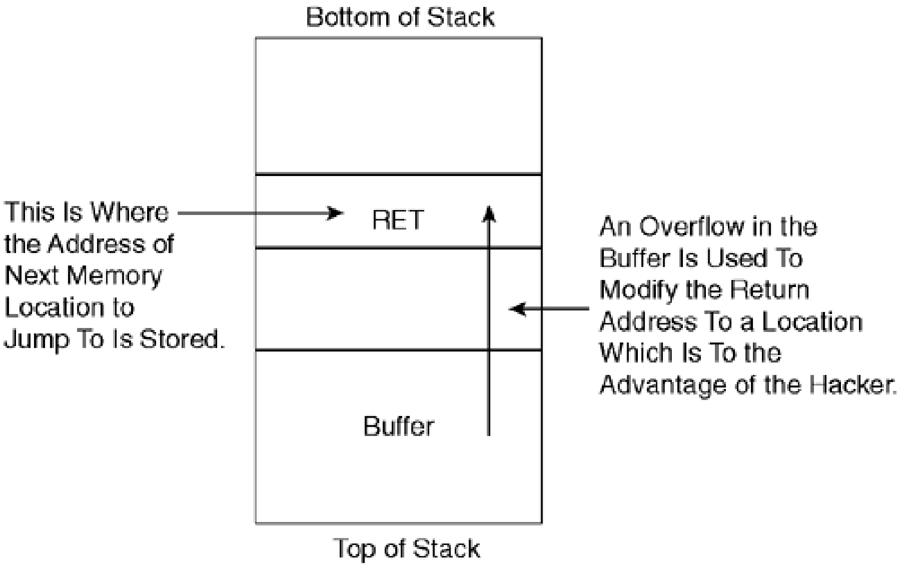

Bottom of Stack
This Is Where the Address of Next Memory Location to Jump To Is Stored.
RET
An Overflow in the Buffer Is Used To Modify the Return Address To a Location Which Is To the Advantage of the Hacker.
Buffer
Top of Stack

Figure 2-4. Buffer Overflow Attack (*smashing the stack)*

NIDS can prevent buffer overflow attacks by looking for data elements in packets that look suspiciously large and that can cause various types of buffer overflows. For example, some NIDS can detect this type of attack by looking for Telnet packets that have an abnormally long New Environment Variable Telnet option.

7. Getadmin Exploits

Getadmin exploits is a popular class in privilege escalation attacks, with a large number of variations and attack strategies available. It's generally carried out using a tool called getadmin.exe.

Getadmin is a program that works by tampering in NTOSKRNL.EXE which is low-level kernels file in Windows NT. By changing one of the bits, and turns off a checking mechanism in the OS that is critical to system security.

Windows NT has security routines, which check whether a user is authorized to start a new process or thread. NT also has a mode that allows users who have debug privileges to execute any threads or processes. When getadmin change bit, it will turn off checking to see whether a user has this type of privilege. So security routines no longer check to see whether a user has the rights to execute a process, thereby giving the hacker many more privileges than the user originally had.

Many getadmin attacks start with an attacker's gaining access through a guest login account, so NIDS must focus on detecting guest login attempts to the NT systems and providing suitable countermeasures based on the security policies.

2.4 Data mining approach as tool in Intrusion Detection /Prevention Systems.

Data mining is becoming one of the popular techniques for detecting intrusion besides other techniques such as expert systems, state transition

analysis, and statistical analysis. Recently, many IDS applications are tending to use different data mining techniques for detecting intrusions. Data mining becomes important in IDS/IPS because it can help in the following:

- Learning from traffic data (Supervised learning): learn precise models from past intrusions and unsupervised learning: identify suspicious activities.
- Maintaining models on dynamic data.
- Correlating suspicious events across network sites helps detect sophisticated attacks not identifiable by single site analyses.
- Analyzing long term data (months/years), uncover suspicious stealth activities (e.g. insiders leaking/modifying information).

2.4.1 Misuse detection

The feature's characteristics of this type are:

- Building predictive models from labeled data sets (instances are labeled as “normal” or “intrusive”).
- Can only detect known attacks and their variations.
- High accuracy in detecting many kinds of known attacks.
- Look for known indicators ICMP Scans, port scans, connection attempts CPU, RAM I/O Utilization, File system activity, modification of system files, permission modifications. Such as using rule if (src_ip == dst_ip) then “land attack".

Many IDS /IPS systems especially based on a network use combination of data mining approach to develop rules for misuse detection such as:

a. Classification of intrusions

- Using RIPPER algorithm in Network Intrusion Detection [25], ADAM Audit Data Analysis and Mining project at George Mason University is developing anomaly detection algorithms based on automated audit data analysis with depend on Bayesian classifier in Intrusion Detection [26] , fuzzy association rules [27] , decision trees for Network Intrusion Detection [28], neural networks [29] , and genetic algorithms [30].
- Building multiple agents of Classification Models consists of Tree based model, Bayesian model, and Neural Network based model [35].

b. Association pattern analysis

- Building normal profile in ADAM [26], frequent episodes for constructing features [31].

c. Cost sensitive modeling

- Using AdaCost: Misclassification Cost-Sensitive Boosting and MetaCost [32].

d. Learning from rare class

- PNrule is a model consisting of positive rules (P-rules) that predicts presence of the class, and negative rules (N-rules) that

- predict absence of the class. PNrule a new framework using in Network Intrusion Detection [33, 34].

e. MADAM ID

(Mining Audit Data for Automated Models for Intrusion Detection) developed at Columbia University uses data mining techniques to discover patterns of intrusions, and this system consists of classification and meta-classification to learn the signature of attacks, association rules algorithm determines relationships between fields in the audit trail records and frequent episodes algorithm models sequential patterns of audit events. This system was previously known as *JAM* (Java Agents for Meta learning), before developing it by building additional components [36, 37].

f. IOWA-ADCPRID

(IOWA-Automated Discovery of Concise Predictive Rules for Intrusion Detection) System develops at Iowa State University [38]. This system implements data mining to provide global and temporal views of intrusions on a distributed system by using a genetic algorithm selects feature subsets to reduce the number of observed features while improving learning accuracy.

The rules detect intrusions against programs using feature vectors to describe the system calls executed by each process.

2.4.2 Anomaly detection

While this type is characterized by features

- Baseline the normal traffic and then look for things that are out of this behavior will detect it as attacks.

- Relatively high false positive rate anomalies can just be new normal activities such as system behaviors may be recognized as anomalies.

The IDS /IPS systems that use data mining approach for Anomaly detection:

a. Statistical approaches
   - Using Finite mixture model which is on-line unsupervised learning of a probabilistic model detects outliers in an online process for network intrusion detection system [39].

b. Various anomaly detection
   - Instance Based Learning (IBL) model in which query data is classified according to its relation to a set of previously encountered exemplar instances. The system stores historical examples of user behavior to reference when assessing the normalcy of newly encountered behavioral data. This approach use reduces an anomaly detection problem and data reduction for intrusion detection system [40].

- NNID (Neural Network Intrusion Detector) is characterized by use the patterns of behavior can be learned for behavior of an individual user is an easier task than trying to do it for all users simultaneously and detecting an anomaly intrusion in real-time [41]. Use similarity trees (decision trees with Boolean logic functions) to profile each legitimate customer's behavior to detect deviations from the normal and cluster analysis to separate each legitimate customer's credit card transactions. [42].

- Use Distribution Based Artificial Anomaly (DBA2) Generation Algorithm for artificial anomalies to Detect Unknown and Known Network Intrusions [43].
- Use model consist of Density-based and grid-based clustering algorithm that is suitable for unsupervised anomaly detection [44].

c. Outlier detection schemes

- Use Nearest neighbor approaches for Machine Learning Approach to Anomaly Detection [45].
- Density based by using degree is called the local outlier factor (LOF) of an object to find local outliers [46].
- Use Clustering in build machine learning model [47].

d. IDDM

(Intrusion Detection using Data Mining) project aims to explore data mining as a supporting paradigm in extending intrusion detection capabilities and reduce anomaly detection problem using combine of techniques: Classification, Association, Frequent Episodes, Clustering and Meta-rules [48].

e. ADAM

(Audit Data Analysis and Mining) is a system for using data mining techniques to detect intrusions. It uses a combination of association rules mining and Decision-tree classifier to mining unexpected rules in a network audit trails and discovers an attack [49, 50].

f. MINDS

(Minnesota Intrusion Detection System) is a system which uses a suite of data mining techniques to automatically detect attacks against computer networks and systems. It's developed and used by the University of Minnesota. This system provide an anomaly detection technique that assigns a score to each network connection that reflects how anomalous the connection is, and an association pattern analysis based module that summarizes those network connections that are ranked highly anomalous by the anomaly detection module [51].

2.5. Related work "Application IDS/IPS Using both Misuse and Anomaly Detection"

Much research and application being to combine misuse and anomaly intrusion detection techniques and use data mining approach to achieve this uniform system for capable detecting both known and unknown intrusions.

IIDS

Intelligent Intrusion Detection System Architecture [52] is distributed and network-based modular architecture to monitor activities across the whole network. In IIDS multiple sensors, both anomaly and misuse detection sensors serving as experts, information from different intrusion detection sensors using in Decision Engine consist of Fuzzy Cognitive Maps (FCMs) and fuzzy rule-bases. The IIDS architecture runs in a high speed cluster environment. In this environment, the Decision Engine resides in the head node and monitors intrusion activities across cluster. This system on-line process and also suffer from false alarm in some cases.

RIDS-100

RIDS Rising Intrusion Detection System (RIDS) [53] is provided by Rising Tech. RIDS makes the use of both intrusion detection technique, misuse and anomaly detection. Distance based outlier detection algorithm used for anomaly detection while misuse detection can be matched data pattern with scanned network data using data mining classification Decision Tree algorithm. It is Commercial system under development to cover all possibilities of anomaly detection problems and complexity.

Snort

It is an open source using in a network intrusion prevention and detection system (IDS/IPS) developed by Sourcefire [54]. It’s development to combining the benefits of misuse detection, and anomaly-based detection. Packet after sniffer by snort goes processing through the following steps: decoding header information of packet at the different layers, preprocessor functions such as IP fragment or TCP stream reassembly, detect by evaluation of a subset of rules according to the information packet that captured, finally alert if a match is found, the corresponding action is carried out. But still limited in anomaly detection and sometime not take any action against unusual amounts of network traffic.

ESIDE-Depian (Intelligent Security Environment for Detection and Prevention of Network Intrusions) [16], presents the first unified misuse and anomaly Detection system based on Bayesian Networks to analyze completely network packets, and the strategy to create a consistent knowledge model that integrates misuse and anomaly-based knowledge. Using integrates snort as misuse detector trainer so the Bayesian Network of five experts is able to react against both Misuse and Anomalies.

Chapter three:
proposed system Methodology

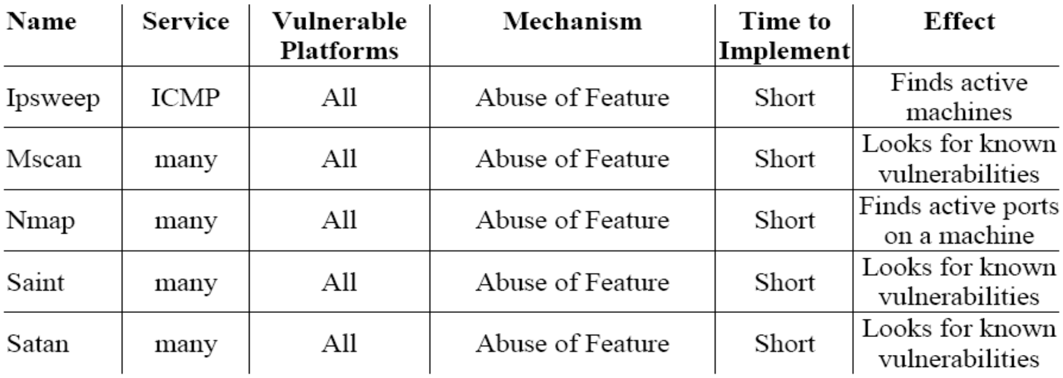
3.1. Introduction

The strength of any network security system lies on its weakest point. Most of the protection systems either a prevention type or detection, focuses on the protection borders of the network from external attacks.

The dangerous attacker on a network will be an insider as well as a professional hacker. The reason for that is most network policies are not very stringent in defining rules and codes of behavior for users on the internal network. In addition, everyone on an inside network is generally trusted. This can allow this user to launch any of the attacks, with devastating results.

The important type of activity that often precedes all network attacks is reconnaissance or probing, which is in some research classified as type of attack. Unfortunately, from the perspective of someone performing a scan, they are legally searching the network to find publicly accessible resources, especially if it has been done from inside network.

The attacker has several methods that can be used either to hide the fact that a probe is occurring, or obscure the location of the party who is performing the probe. These methods make it difficult to detect a probe by NIPS [56]. Figure 3.1 provides a summary of some probes tool.



Name	Service	Vulnerable Platforms	Mechanism	Time to Implement	Effect
Ipsweep	ICMP	All	Abuse of Feature	Short	Finds active machines
Mscan	many	All	Abuse of Feature	Short	Looks for known vulnerabilities
Nmap	many	All	Abuse of Feature	Short	Finds active ports on a machine
Saint	many	All	Abuse of Feature	Short	Looks for known vulnerabilities
Satan	many	All	Abuse of Feature	Short	Looks for known vulnerabilities

Figure 3.1 summary of the probes [56]

The factors listed bellow makes it possibly hard to any Detections/Prevention tools to detect any possible "Probe" attack:

- Scan lingeringly and randomly: one of the methods of scanning stealthily is probing tool can be configured to occur slow and probe ports or machines in a nonlinear order. In general An IPSs / IDSs will have a very hard time identifying one stray connection per hour to a random port as a port sweep initiated by an attacker.

- Probe with Half-Open or Other Unlogged Connections: Another method an attacker used is to hide the fact a probe is happening to be probe with half-open connections. A connection for which the three way TCP handshake is never completed will not be logged by the operating system.

- Use an Intermediate Machine to Hide the Real Source of the Scan: One method attackers can obscure their identity is to use an ftp (file transfer protocol) bounce probe.

- The reason some ftp servers will allow anyone to tell them to send data to a particular port on a particular machine. An attacker can look at the response the ftp server gives from such a request and determines whether that port is listening on the victim machine. The portscan will appear to be coming from an anonymous ftp server, and this simple step may be enough to assure that the party who is really doing the scanning is never identified.

Although it can be impossible to protect against all types of attacks, such an individual can implement, the improved NIPS based on both misuse and anomaly approach can expose such an individual and lead to prevent before more damage is done.

To achieve this objective to be the NIPS performs requirements such as detect probe attack and prevent it before launch network attack to the target machine with high performance, detect scans either from an outside or inside source of network, reduce false alarm, easy implementing with low cast, and

compatibility with any operating system. This chapter shows the proposed system design, and described in details the tool used in building up the system.

3.2. The Proposed System Design

The System consists of four main phases as described in Figure 3.2 bellow. Each phase performs a certain task. These phases integrate with each other to manufacture the complete system, eventually. An appropriate technology was used in each phase when building the system, in order to meet its requirement.

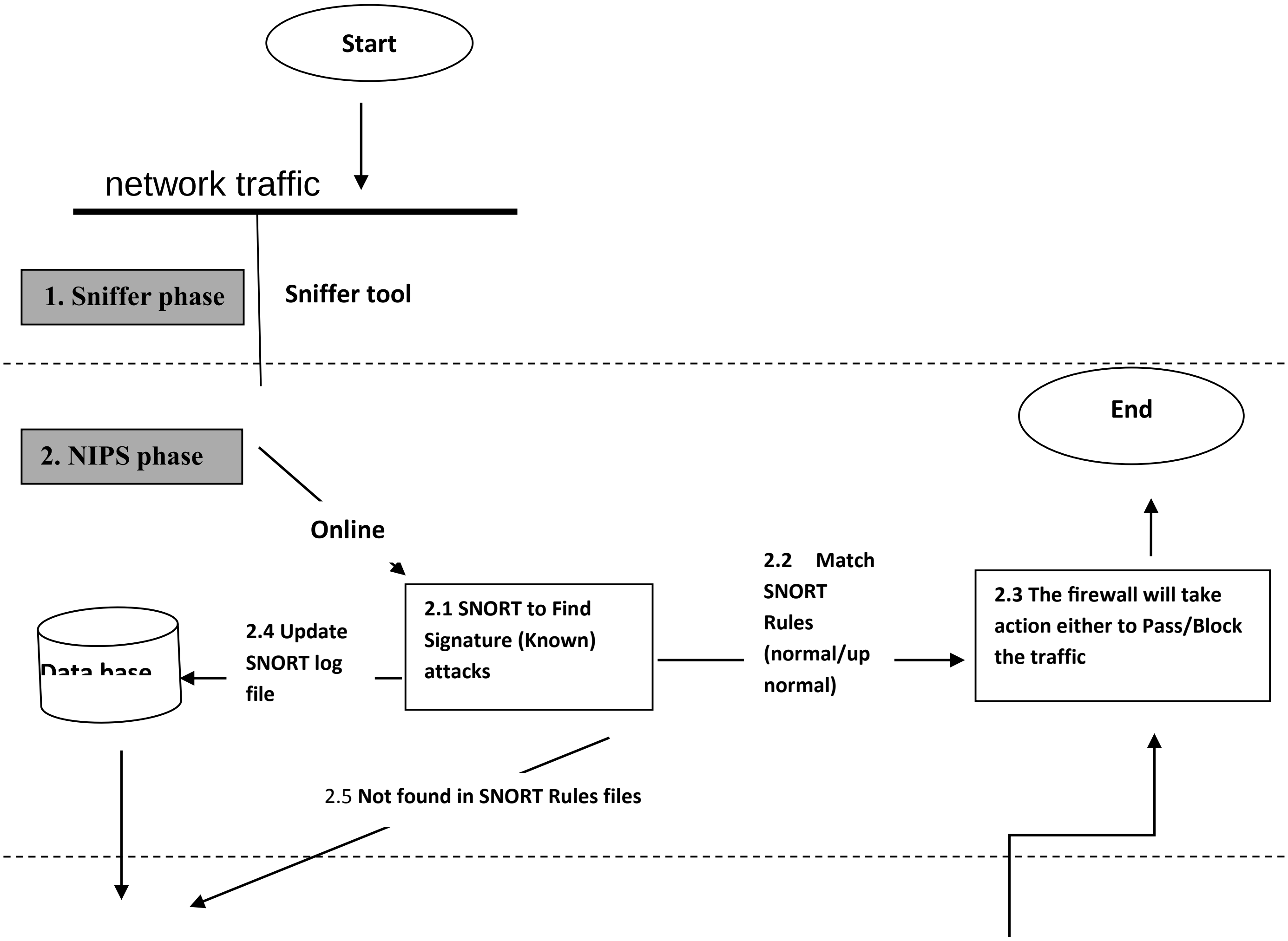
1. The Sniffer phase: The First phase started with packet sniffer by network sniffer tools, which are capturing packets from the network with different levels of detail and displays it on the control unit in the next phase to examine data from a live network or from a capture file on disk.

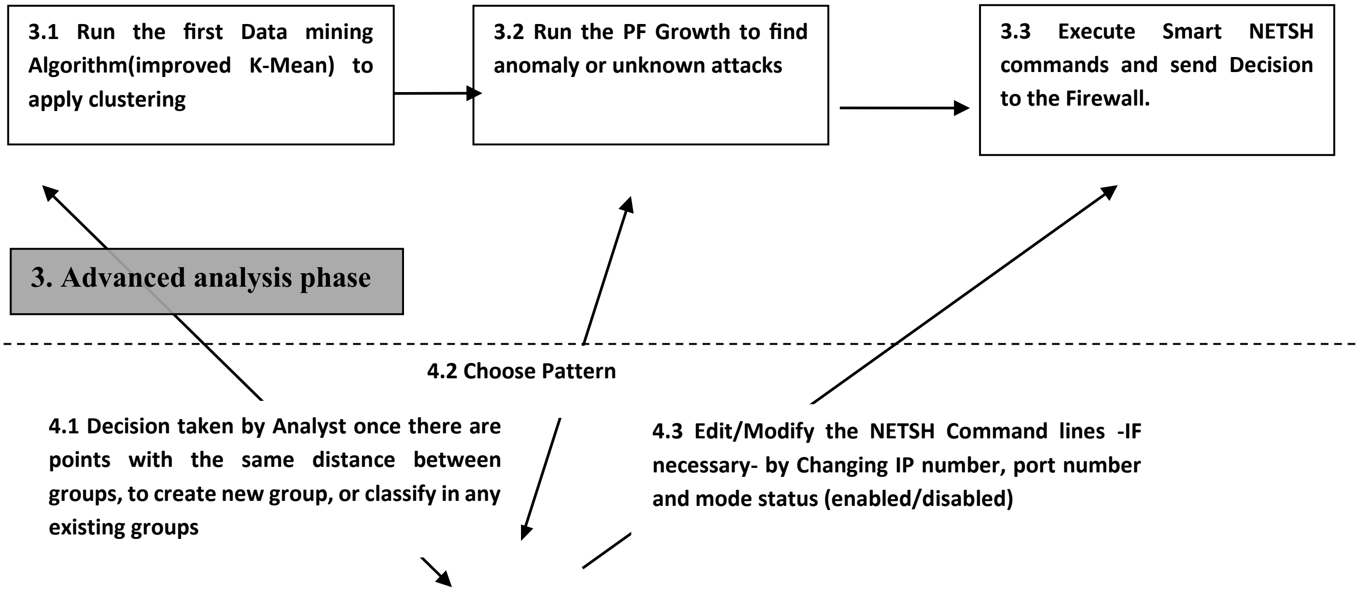
2. Then followed by NIPS phase that have several actions started with analysis of the captured packet, take the appropriate decision such as pass, drop ,and log it into a file(Stored the log file in Offline DB) based on a set of rules defined in SNORT tools., and finally to give an alert in an analyst user interface with action intended for target traffic. In addition the updated database of log files (Steps from 2.1 to 2.5 in figure 3.2 describe the process in details).

3. Then begin phase of an advanced analysis to a database of log file. This stage consisted of two sub-phases of data mining approach. One Sub-phase is clustering based method using suggestion improve K-mean algorithm by partitioning data into clusters of a similar object and unsupervised learning process of hidden data. Other sub-phase association rules use PF-growth algorithm sought to obtain the frequent pattern for large database. This phase generated new rules help to improve defined rule that used in previous phase (Steps from 3.1 to 3.3 in figure 3.2 describe the process in details).

4. Control Analyst Interface is a final phase to choose recent rules, which suit with anomaly detection. In addition take the suitable decision about

5. the new rules, which lies in the middle of clustering. As well as update proposal smart new rule file will be searched in it in case still not normal traffic or suspicious to appropriate actions (Steps from 4.1 to 4.3 in figure 3.2 describe the process in details).


Start
network traffic
1. Sniffer phase
Sniffer tool
2. NIPS phase
End
Online
2.2 Match SNORT Rules (normal/up normal)
2.3 The firewall will take action either to Pass/Block the traffic
Data base
2.4 Update SNORT log file
2.1 SNORT to Find Signature (Known) attacks
2.5 Not found in SNORT Rules files



3.1 Run the first Data mining Algorithm(improved K-Mean) to apply clustering

3.2 Run the PF Growth to find anomaly or unknown attacks

3.3 Execute Smart NETSH commands and send Decision to the Firewall.

3. Advanced analysis phase

4.2 Choose Pattern

4.1 Decision taken by Analyst once there are points with the same distance between groups, to create new group, or classify in any existing groups

4.3 Edit/Modify the NETSH Command lines -IF necessary- by Changing IP number, port number and mode status (enabled/disabled)

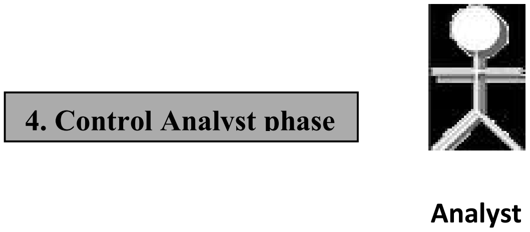


Figure 3.2: Scheme of the proposed system

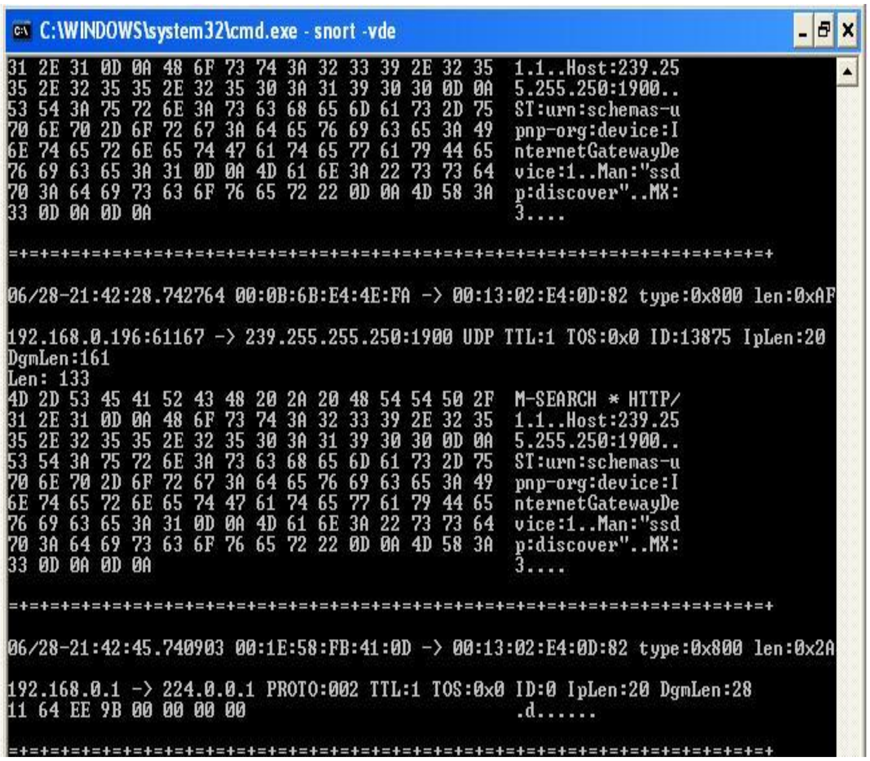
3.3. The Used Tools.

Now, the tools and algorithm used in each phase to improve the system are being discussed in details in this section.

3.1.1Sniffer phase tool

There are many sniffer tools, which can be defined as either device or program that allows monitoring the traffic traveling between networked computers. The packet sniffer tool installed on the "network server" will capture data related to other machines inside the network in order to save it for later analysis.

One of the packet sniffer tools that is used for network troubleshooting and other useful purposes is Tcpdump. It is classic IP sniffer for network monitoring and data acquisition. It can be used to print out the headers of packets on a network interface that matches a given expression. A Winpcap module is composed by the packet capture that runs at the kernel level, while the packet.dll and Libpcap



```
31 2E 31 0D 0A 48 6F 73 74 3A 32 33 39 2E 32 35  1.1..Host:239.25
35 2E 32 35 35 2E 32 35 30 3A 31 39 30 30 0D 0A  5.255.250:1900..
53 54 3A 75 72 6E 3A 73 63 68 65 6D 61 73 2D 75  ST:urn:schemas-u
70 6E 70 2D 6F 72 67 3A 64 65 76 69 63 65 3A 49  pnp-org:device:I
6E 74 65 72 6E 65 74 47 61 74 65 77 61 79 44 65  nternetGatewayDe
76 69 63 65 3A 31 0D 0A 4D 61 6E 3A 22 73 73 64  vice:1..Man:"ssd
70 3A 64 69 73 63 6F 76 65 72 22 0D 0A 4D 58 3A  p:discover"..MX:
33 0D 0A 0D 0A                                   3....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/28-21:42:28.742764 00:0B:6B:E4:4E:FA -> 00:13:02:E4:0D:82 type:0x800 len:0xAF

192.168.0.196:61167 -> 239.255.255.250:1900 UDP TTL:1 TOS:0x0 ID:13875 IpLen:20
DgmLen:161
Len: 133
4D 2D 53 45 41 52 43 48 20 2A 20 48 54 54 50 2F  M-SEARCH * HTTP/
31 2E 31 0D 0A 48 6F 73 74 3A 32 33 39 2E 32 35  1.1..Host:239.25
35 2E 32 35 35 2E 32 35 30 3A 31 39 30 30 0D 0A  5.255.250:1900..
53 54 3A 75 72 6E 3A 73 63 68 65 6D 61 73 2D 75  ST:urn:schemas-u
70 6E 70 2D 6F 72 67 3A 64 65 76 69 63 65 3A 49  pnp-org:device:I
6E 74 65 72 6E 65 74 47 61 74 65 77 61 79 44 65  nternetGatewayDe
76 69 63 65 3A 31 0D 0A 4D 61 6E 3A 22 73 73 64  vice:1..Man:"ssd
70 3A 64 69 73 63 6F 76 65 72 22 0D 0A 4D 58 3A  p:discover"..MX:
33 0D 0A 0D 0A                                   3....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/28-21:42:45.740903 00:1E:58:FB:41:0D -> 00:13:02:E4:0D:82 type:0x800 len:0x2A

192.168.0.1 -> 224.0.0.1 PROTO:002 TTL:1 TOS:0x0 ID:0 IpLen:20 DgmLen:28
11 64 EE 9B 00 00 00 00                          .d......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

(library of Winpcap) that run at a user level in order to track down network problems or to monitor network activities as described in Figure 3.3. Tcpdump is the source of the Libpcap/WinPcap packet capture library.

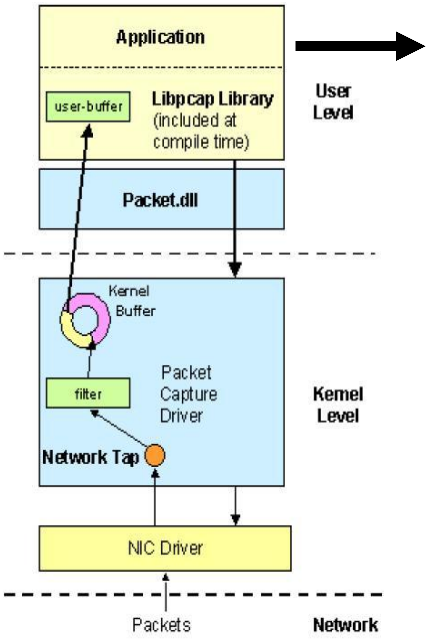


Figure 3.3: the capturing packet

Windows Packet Capture (WinPcap) is an application programming for capturing network traffic and filtering engine of many open source and monitoring network tools including protocol analyzers, network monitors, network intrusion detection/prevention systems, sniffers, traffic generators and network testers [57].

WinPcap consists of drivers for the Windows system which enables applications to send and receive raw network packets to/ from a network adapter directly. Receiving raw network packets is also known as packet capturing, therefore the name "Windows Packet capture library".

WinPcap support saves captured packets to a file, and reading files containing saved packets; applications can be written, using WinPcap, to be able to capture network traffic and analyze it, or to read a saved capture and analyze it, using the same analysis code.

A capture file saved in the format that WinPcap uses can be read by applications that understand that format usually it is an XML file. The WinPcap API writes it in C, so other languages such as Java, .NET languages, and scripting languages generally use a wrapper.

This tool is capable of capturing and transmitting network packets by going around the protocol stack, filtering network packets, and generating network statistics.

3.1.2NIPS phase tool

Snort is an open source of a network intrusion prevention and detection system. It uses a rule-based language combining signature, protocol and anomaly inspection methods.

Snort is a very powerful tool and is known to be one of the best IDS on the market even when compared to commercial IDS. Snort is written by Martin Roesch. It was bought by the commercial company SourceFire which was bought itself by the FireWall Giant CheckPoint in 2005.

It is designed primarily for small network segments and very flexible due to its rule-based architecture. Snort is very easy to insert and expand upon rules as new security threats are detected. Snort uses the libpcap library by WinPcap tool to capture packets then uses new rule types to help pass or drop packets based on Snort rules [58].

The Snort construction consists of three primary components, which are Packet Decoder, Detection Engine, and Logger/Alerter.

Packet decoder performs all the work to organize the data in an appropriate way for the detection engine. Detection Engine, which presents the heart of Snort. It is responsible for inspecting every single packet based on the Snort rules file that is loaded at runtime. It applies rules to packets if the rule matches the decoded packet then triggers the action specified in the rule definition. Otherwise a packet that does not match any snort rule set is simply discarded [59].

Detection Engine works on inspected IP header of the packet in a network layer, the transport layer header, the application layer level header, packet content, and size of packet payload. The final output process generates alerts to a user and logs event saved in the database. As is shown in the Figure 3.4

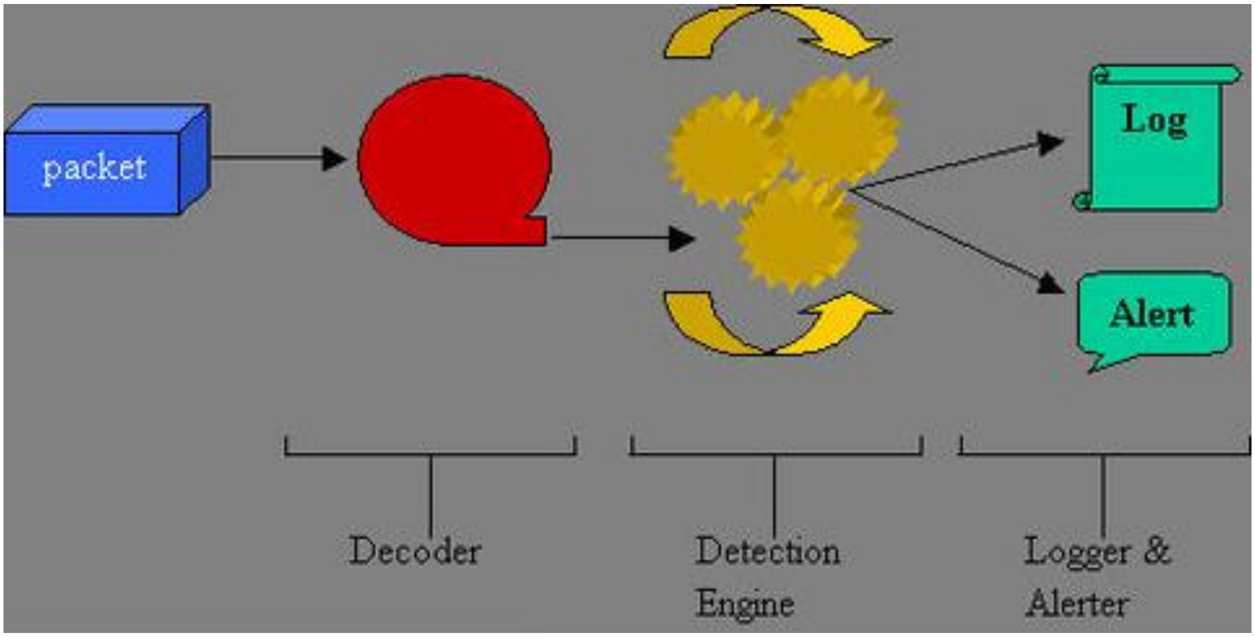


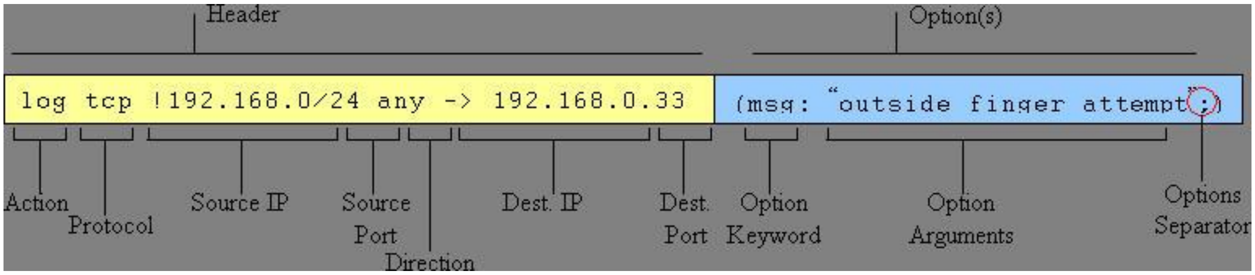
Figure 3.4 Snort Architecture

Snort can be run in four modes:

- Sniffer mode: snort will read the network traffic and print them to the screen.
- Packet logger mode: snort will record the network traffic on a file.
- IDS mode: network traffic matching security rules will be recorded.
- IPS Intrusion prevention system mode: also known as snort-inline.

The Snort rules file is an ASCII file that can be created and edited using any editor tool. These should be consistent with the overall Intrusion detection policy. Snort rules must be specified on a single line, because that the Snort rule analyst doesn't know how to handle rules on multiple lines.

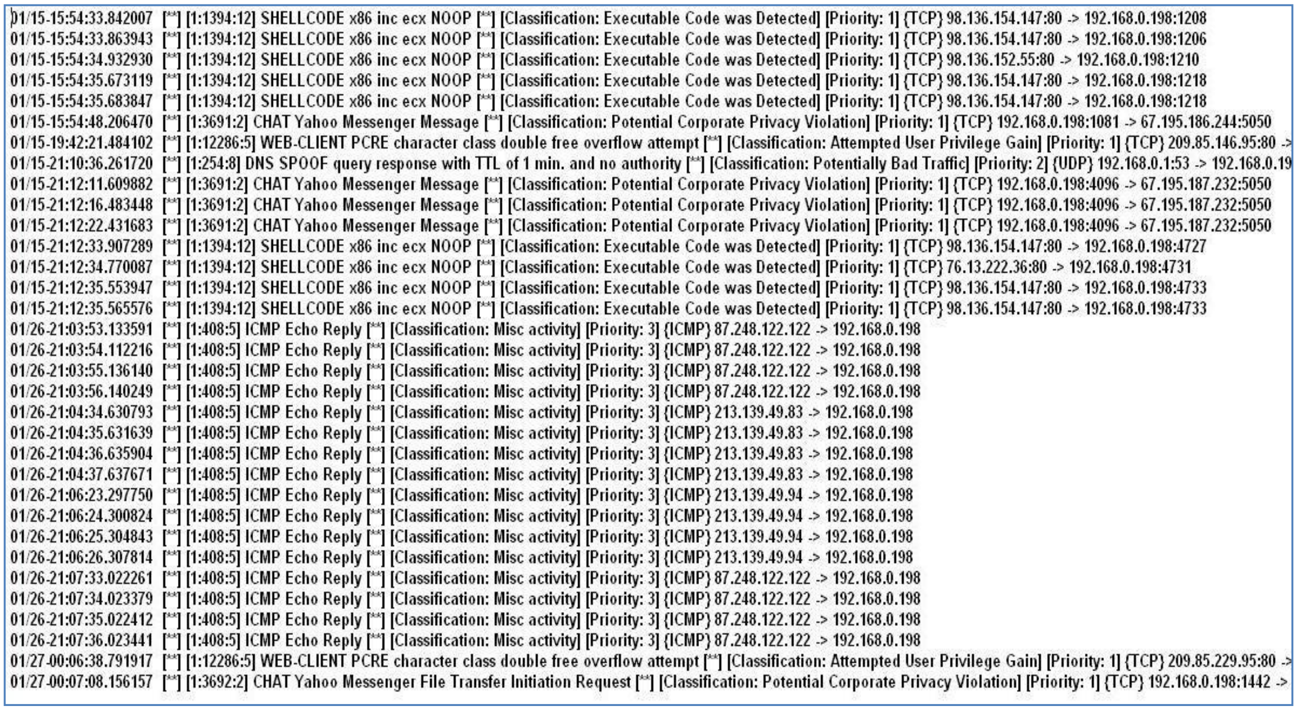
Snort rules consist of two logical parts: the rule header and options. The first part the “Rule header” contains the rule's action (e.g. drop, reset, and log), protocol, source and destination IP addresses and net-masks, and the source and destination ports information. While second part; the “Rule option” contains alert messages and information on which parts of the packet should be inspected, therefore to determine what the rule action should be taken. As shown in the Figure 3.5



```
log tcp !192.168.0/24 any -> 192.168.0.33 (msg: “outside finger attempt”;)
```

Figure 3.5 the Snort rule header and options details

The logger of snort can be saved in a text file or in a database. The most important contents of the log file that are used as input in the Data mining algorithm (K-mean) are date, time of entry, the IP and port of source, and the IP and port of destination, in addition to protocol, classification of rules, priorities of security, and information about the traffic as shown in the Figure 3.6. In the proposed system Microsoft SQL Server 2008 is used to save the detail of log file in the database. The database content is used in advanced analysis phase.



```
01/15-15:54:33.842007  [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 98.136.154.147:80 -> 192.168.0.198:1208
01/15-15:54:33.863943  [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 98.136.154.147:80 -> 192.168.0.198:1206
01/15-15:54:34.932930  [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 98.136.152.55:80 -> 192.168.0.198:1210
01/15-15:54:35.673119  [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 98.136.154.147:80 -> 192.168.0.198:1218
01/15-15:54:35.683847  [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 98.136.154.147:80 -> 192.168.0.198:1218
01/15-15:54:48.206470  [**] [1:3691:2] CHAT Yahoo Messenger Message [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {TCP} 192.168.0.198:1081 -> 67.195.186.244:5050
01/15-19:42:21.484102  [**] [1:12286:5] WEB-CLIENT PCRE character class double free overflow attempt [**] [Classification: Attempted User Privilege Gain] [Priority: 1] {TCP} 209.85.146.95:80 ->
01/15-21:10:36.261720  [**] [1:254:8] DNS SPOOF query response with TTL of 1 min. and no authority [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 192.168.0.1:53 -> 192.168.0.19
01/15-21:12:11.609882  [**] [1:3691:2] CHAT Yahoo Messenger Message [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {TCP} 192.168.0.198:4096 -> 67.195.187.232:5050
01/15-21:12:16.483448  [**] [1:3691:2] CHAT Yahoo Messenger Message [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {TCP} 192.168.0.198:4096 -> 67.195.187.232:5050
01/15-21:12:22.431683  [**] [1:3691:2] CHAT Yahoo Messenger Message [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {TCP} 192.168.0.198:4096 -> 67.195.187.232:5050
01/15-21:12:33.907289  [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 98.136.154.147:80 -> 192.168.0.198:4727
01/15-21:12:34.770087  [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 76.13.222.36:80 -> 192.168.0.198:4731
01/15-21:12:35.553947  [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 98.136.154.147:80 -> 192.168.0.198:4733
01/15-21:12:35.565576  [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 98.136.154.147:80 -> 192.168.0.198:4733
01/26-21:03:53.133591  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 87.248.122.122 -> 192.168.0.198
01/26-21:03:54.112216  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 87.248.122.122 -> 192.168.0.198
01/26-21:03:55.136140  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 87.248.122.122 -> 192.168.0.198
01/26-21:03:56.140249  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 87.248.122.122 -> 192.168.0.198
01/26-21:04:34.630793  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 213.139.49.83 -> 192.168.0.198
01/26-21:04:35.631639  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 213.139.49.83 -> 192.168.0.198
01/26-21:04:36.635904  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 213.139.49.83 -> 192.168.0.198
01/26-21:04:37.637671  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 213.139.49.83 -> 192.168.0.198
01/26-21:06:23.297750  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 213.139.49.94 -> 192.168.0.198
01/26-21:06:24.300824  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 213.139.49.94 -> 192.168.0.198
01/26-21:06:25.304843  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 213.139.49.94 -> 192.168.0.198
01/26-21:06:26.307814  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 213.139.49.94 -> 192.168.0.198
01/26-21:07:33.022261  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 87.248.122.122 -> 192.168.0.198
01/26-21:07:34.023379  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 87.248.122.122 -> 192.168.0.198
01/26-21:07:35.022412  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 87.248.122.122 -> 192.168.0.198
01/26-21:07:36.023441  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 87.248.122.122 -> 192.168.0.198
01/27-00:06:38.791917  [**] [1:12286:5] WEB-CLIENT PCRE character class double free overflow attempt [**] [Classification: Attempted User Privilege Gain] [Priority: 1] {TCP} 209.85.229.95:80 ->
01/27-00:07:08.156157  [**] [1:3692:2] CHAT Yahoo Messenger File Transfer Initiation Request [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {TCP} 192.168.0.198:1442 ->
```

Figure 3.6 Details Log File of the Snort

However, the Snort tool faces some challenges. In Misuse detection, rules database is large and continues to grow. In addition, Snort spends 80% work time to do string match. While the probability of identifying new an attack is low in anomaly detection. Snort is unable to detect internal network scans because it does not examine them like most of the existing intrusion detection systems.

Some new attempts to improve snort tool is by increasing the ability of the detect engine, or using a hybrid architecture of hardware to reduce the workload, in addition to use better detection algorithms.

3.3.3 phase of an advanced analysis tool

As mentioned earlier, despite the power of the Snort, this program requires some improvements to overcome its weaknesses. One of the suggestions is using data mining algorithms because the work with a huge database such as a snort logger file , in addition it can help an analyst to discover new rules from a hidden pattern that snort tool can't recognize it as obvious rules; thus improves the work

The proposed advanced analysis phase consists of two sub-phases of data mining approach. The first phase is clustering based method using suggested improved K-mean algorithm by partitioning data into clusters of a similar object and unsupervised learning process of hidden data. The input to this sub-phase is a log file of snort saved in a database. The elements of the input file will be focused on it and used within this algorithm is the IP and port of source, and the IP and port of destination. The output of this sub-phase is group of clusters that represent a normal traffic cluster, worm cluster, probe (scan) cluster, and new discover cluster that is named and generated by a decision of Control Analyst phase.

Using improved K-mean algorithm leads to reduce time process by selecting manually a center point for each purposed cluster and a center point of recent discovered cluster will be a first input point that is far from existing clusters or midpoint between clusters depend on new decision of Control Analyst phase or history decision if repetition same case.

The manual selection of center points is about selecting a center point by plotting the candidates' points and calculating the mid-point from two corresponding points of the boundary cluster. Then choosing the point closest to the midpoint and considering it as the center point to the cluster. The using of the improved K-mean algorithm as the first sub-phase in the proposed system leads to reduce time process and time consumption compared with original K-means algorithm that selects the center points for each cluster randomly.

Other sub-phase association rules use the FP-growth algorithm (also known as frequent pattern algorithm) sought to obtain the frequent pattern for large database. The input to this sub-phase is records from clusters generated by the previous sub-phase K-mean sub-phase. The output will be a group of pattern for each cluster from the discovered hidden pattern where it is possible to select rules to prevent the IP source from targeting IP destination. These rules can be executed manually by Netsh command-line utility in Windows OS in order to change the configuration of firewall. The Integration between these two sub-phases helps to discover new rules especially those related to internal network scans. In addition to unsupervised learning process in K-mean algorithm used to discover new cluster which may represent new type of attack depending on user decision. In general all interface and the used algorithms will be written and programmed by Microsoft Visual Basic 2008 (VB 9.0).

3.3.4.Control Analyst Interface tool

Microsoft Visual Basic 2008 (VB 9.0) or Visual Basic .NET, is an object-oriented computer programming language, which is implemented on the .NET Framework. Microsoft currently supplies two major implementations of Visual Basic: Microsoft Visual Studio, and Microsoft Visual Studio Express.

Microsoft added many features in this version, including:

A true conditional operator, "IIf (condition as boolean, truepart, falsepart)", to replace the "IIf" function, Anonymous types, Support for LINQ, Lambda expressions, XML Literals, Type Inference, Extension methods.

Using VB.NET in build Control Analyst Interface and algorithms of sub-phases in pervious phase because maximize the performance, reliability, multi-language application execution environment, scalability, easy use and handle, and security of applications.

3.4.Improved K-means clustering algorithm

K-means clustering is a method of cluster analysis which aims to partition n elements into k clusters in which each element belongs to the cluster with the nearest mean. This method is used in statistics and data mining.

The basic idea of original K-means algorithm is taking K as a parameter represented number of clusters and chooses randomly the initial center point of each cluster. Next step examines each one of the remaining points in the population and assigns it to one of the clusters depending on the minimum distance between center point and other points. The center point is recalculated every time a point is added to the cluster, and this continues until all the points are grouped into the final required number of clusters [3].

This algorithm has time limitation consumption of computation center point and dependence on an algorithm to select randomly of initial center point.

One of the improvements to original K-means algorithm is using K- Medoids algorithm and triangle trilateral relations theorem by selected group of a candidate to be a center point and examines an optimum centre point by minimum distance between these candidate center points, then examines with other rest points [60].

However, this improvement has time consumption because of computation of the optimum centre point twice among candidates of center point and recalculated with other rest points.

The new suggestion, which is suitable with the case of the proposed system, is not selecting a random center point for each cluster. However; it selects center point by plotting candidates of points and calculating midpoint from two corresponded points of the boundary cluster.

Then choosing the point closest to the midpoint and considering it as the center point to the cluster.

The reason behind using the K-Mean even when we choose the initial center point, is that we might need to choose another center points (recalculate new center points).

Steps of Improved proposed of K-means clustering algorithm

```
Start

1. Input:  the number of clusters (k); data object (n).
2. Initialize the new (chosen) centre points sets { $C_1$,$C_{2,\ldots}$ $C_k$ },where j={1,2,…k} .
3. Now using data object $X_{t,}$ t={1,2,…n} and assign each $X_t$ to nearest $C_k$ base on min(D), using distance formula between two points , D=| $x_t$ - $C_j$ $|^2$ .
4. Calculate the distance between two cluster center, d($C_{i,}$ $C_j$ ) ,i=1,2..n, j=1,2,…n
5. Repeat *
6. For each input point $X_{t,}$ t={1,2,..n}
       Begin
         • Set $X_t$ is in the cluster $W_i$ of the centre $C_{i\,;}$
         • If d($C_{i,}$ $C_j$ )>= d($X_{t,}$ $C_i$ ),then $X_t$ is still in current cluster and
           d($X_{t,}$ $C_j$ ), is unchanged ,else calculate distance again;
         • If d($X_{t,}$ $C_i$ )< d($X_{t,}$ $C_j$) assign $X_t$ to $W_j$ temporarily ;
         • calculate min distance($D_t$ ) ,store the value of $D^2_t$
         End
7.  For each cluster $C_j$
    {
       • Calculate the sum of minimum distances squares $E_{j;}$
       • Calculate total $E_{j\,;}$
       •  Update all the current centroid  point sample in $C_j$
       }
```

* Repeat until E is minimal one or no change happen in members of cluster.

8. End

Where:

K: Number of initial clusters, minimum K=2.

n: Number of the clusters data objects, where it depends on the number of attributes in the Snort Log file.

D: Minimum distance between two points

X: The Input point

C: Center point

W: The Selected cluster

E: Minimum distance to all clusters

Note: The main difference between the above K-mean algorithm and the traditional K-Mean algorithm is that the selection of the initial center points is not done randomly anymore.

3.5.FP-Growth Algorithm

In data mining, association rule learning is a popular and well researched method for discovering interesting and hidden relations between items in a large database. Many algorithms for generating association rules were presented over time. Some well known algorithms are Apriori, and FP-Growth, they are algorithms for mining frequent itemsets and generate rules from frequent itemsets found in a database.

Apriori: generate-and-test approach generates candidate itemsets and tests if they are frequent however the generation of candidate itemsets is expensive. While FP-growth is the first algorithm that allows frequent itemset discovery without candidate itemsets generation [61].

Steps of FP-Growth Algorithm

Start

1. Input data: database consisting of Basic features include source and destination IP addresses, source and destination ports and enter the min support number (as integer usually equal 2).Please refer to figure 3.7.

2. Preprocessing operation: Initial scan the frequencies of the items (support of single element item sets) are determined. All infrequent items which are, all items that appear in fewer transactions than min support number are discarded from the transactions. Please refer to figure 3.7

   In addition, the items in each transaction are sorted, so that they are in descending order their frequency in the database.

3. Initial FP-tree: After all individually infrequent items have been deleted from the transaction database; it is turned into an FP-tree. An FP-tree is basically a prefix tree for the transactions (Please refer to figure 3.7). Do the following:

- The sorted list can easily be turned into an FP-tree with a straightforward recursive procedure: at recursion depth k, k-th item in each transaction is used to split the database into sections, one for each item. For each section a node of the FP-tree is created and labeled with the item corresponding to the section. Each section is then processed recursively, split into subsections, a new layer of nodes (one per subsection) is created etc. Note that in doing so one has to take care that transactions that are only as long as the current recursion depth are handled appropriately, that is, are removed from the section before going into recursion.
- In general it may seem to be more natural to build it by inserting transaction after transaction into an initially empty FP-tree, creating the necessary nodes for each new transaction.

- Since the FP-tree is built top down, the parent is already known when the children are created. Thus, it can be passed down in the recursion, where the parent pointers of the children are set directly. Makes it possible to do without parent-to-child pointers entirely.

- An FP-tree node contains only fields for (1) an item identifier, (2) a counter, (3) a pointer to the parent node, (4) a pointer to the successor node (referring to the same item) and (5) an auxiliary pointer that is used when projecting the FP-tree (next operation).

4. Projected FP-tree :(Please refer to figure 3.7)

   Start form bottom to up (from the leaves to the root)

   For k = fewer frequent item to high frequent item do

   {

   - Copy linked nodes to item K form leaf to root and detached from the original FP-tree as well as use auxiliary pointer to Built tree to item K (conditional FP-tree for K)
   - Update the support counts along the conditional FP-tree to reflect the number of transactions containing.

   - Remove the nodes containing item K and add it to pattern file

   - Remove infrequent items (nodes) from the conditional FP-tree (counts node =1 and less than item K).

   - Use the conditional FP-tree for K to find frequent itemsets.

     {

     Repeat

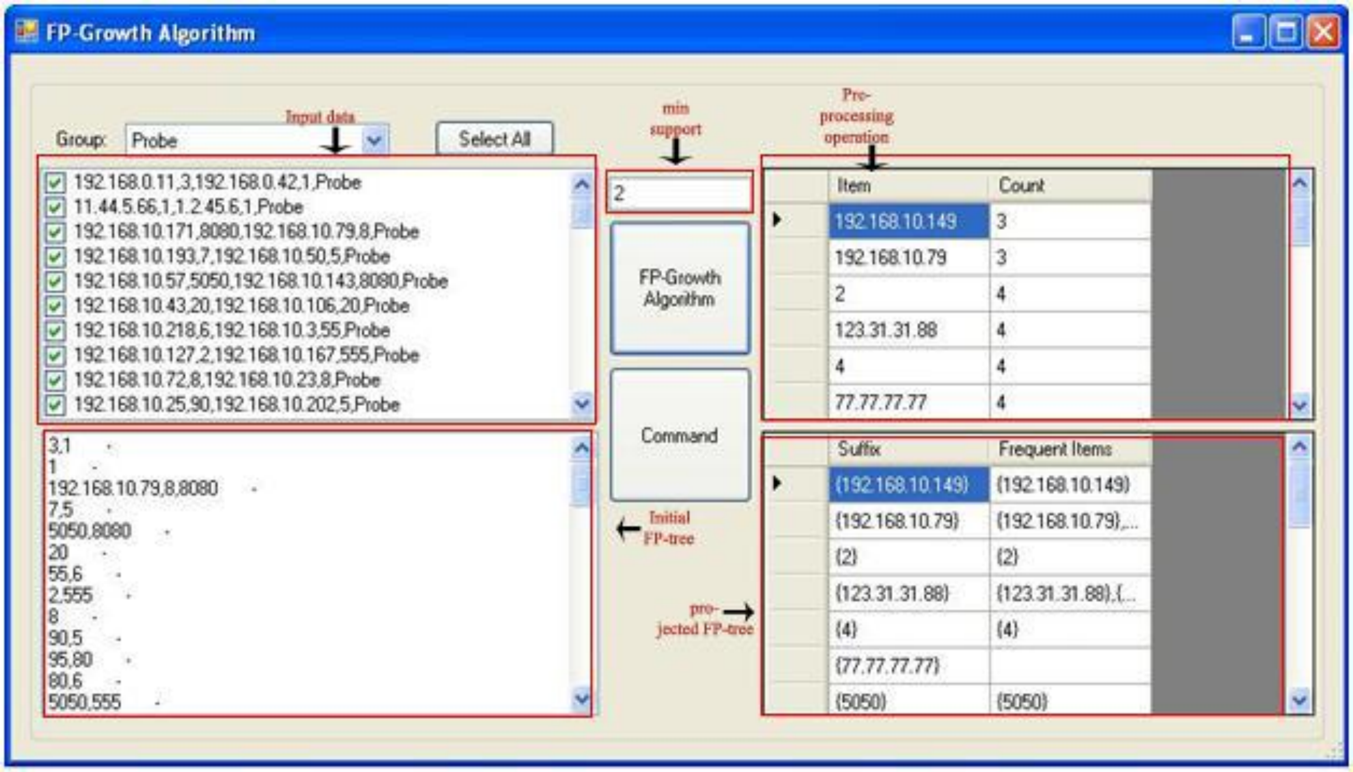
- Read item K from pattern file
- Item K unified with every rest items in conditional FP-tree.
- Stored in the pattern File

} Until: no more frequent itemsets can be extracted, i.e. empty tree or tree with 1 item

}

5. Print all frequent itemsets in file
6. To account for the significance of a pattern, the recall or precision measure can be used. This rule occurred among anomalous connections or this rule occurred in normal connections.

End.



☑ 192.168.0.11,3,192.168.0.42,1,Probe
☑ 11.44.5.66,1,1.2.45.6,1,Probe
☑ 192.168.10.171,8080,192.168.10.79,8,Probe
☑ 192.168.10.193,7,192.168.10.50,5,Probe
☑ 192.168.10.57,5050,192.168.10.143,8080,Probe
☑ 192.168.10.43,20,192.168.10.106,20,Probe
☑ 192.168.10.218,6,192.168.10.3,55,Probe
☑ 192.168.10.127,2,192.168.10.167,555,Probe
☑ 192.168.10.72,8,192.168.10.23,8,Probe
☑ 192.168.10.25,90,192.168.10.202,5,Probe

```
3,1        -
1      -
192.168.10.79,8,8080      -
7,5      -
5050,8080      -
20      -
55,6      -
2,555      -
8      -
90,5      -
95,80      -
80,6      -
5050,555      -
```

2

Item	Count
192.168.10.149	3
192.168.10.79	3
2	4
123.31.31.88	4
4	4
77.77.77.77	4

Suffix	Frequent Items
{192.168.10.149}	{192.168.10.149}
{192.168.10.79}	{192.168.10.79},...
{2}	{2}
{123.31.31.88}	{123.31.31.88},{...
{4}	{4}
{77.77.77.77}	
{5050}	{5050}

Figure 3.7 window of FP-growth in the proposed system

CHAPTER FOUR

Implementation of the Proposed System and Testing

4.1 Introduction

In the previous chapter, the methodological of a proposed system used in building the system in each phase is illustrated. Some of these phases were developed from a collection of several steps depending on a certain technology or tool. In this chapter the implementation of each phase will be explained.

In general, it has begun with the phase of data capture by calling it through the command line of the snort application. Then, analyze this sniffed traffic through the rules' snort that takes the right decision in pass or drops this traffic, and gives the alert in addition to record the events in a log file. After that, it will start the advanced analysis phase on the snort logs file using data mining algorithms for the discovery of hidden behavior of network traffic between the source and destination to find what the snort is overlooked it. Finally, generated candidate rules to improve the performance of snoring and prevent any new attacks in the future.

4.2 Implementation of Sniffer Phase

The Winpcap, when is executed to the window operating systems, it will be ready to use by calling programs. The architecture of Winpcap ability is to communicate and capture data directly from the Network Interface Card, transfers them to the calling programs in order to interpret and processes them depending on program rules, and display results to the user in a comprehensible and productive way. A Winpcap module is composed by the packet capture that runs at the kernel level, while the packet.dll and Libpcap (library of Winpcap) that run at a user level.

In the proposed system, Winpcap was used for NIPS, which is represented by snort application that gives instructions to the Winpcap to start capturing data. According to a selected command line of the snort, the response from Winpcap tool will be based on the used command. As shown in Figure 4.1.

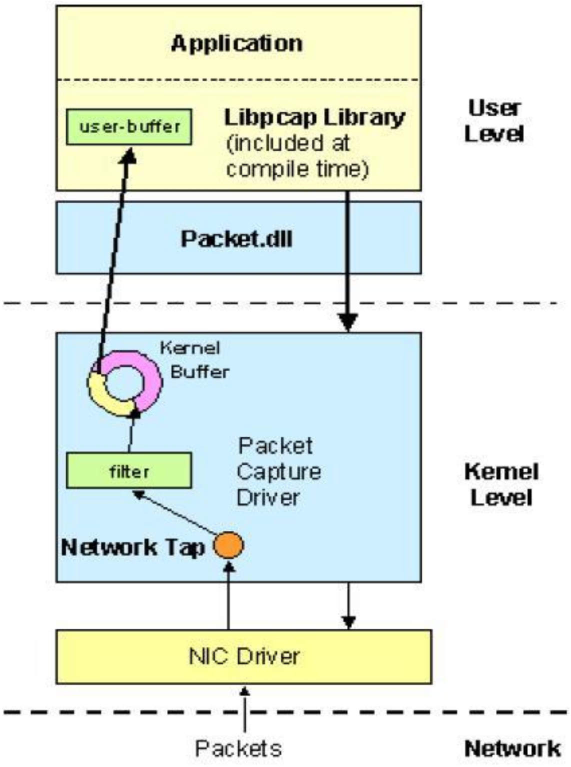


Figure 4.1 structure of the Winpcap

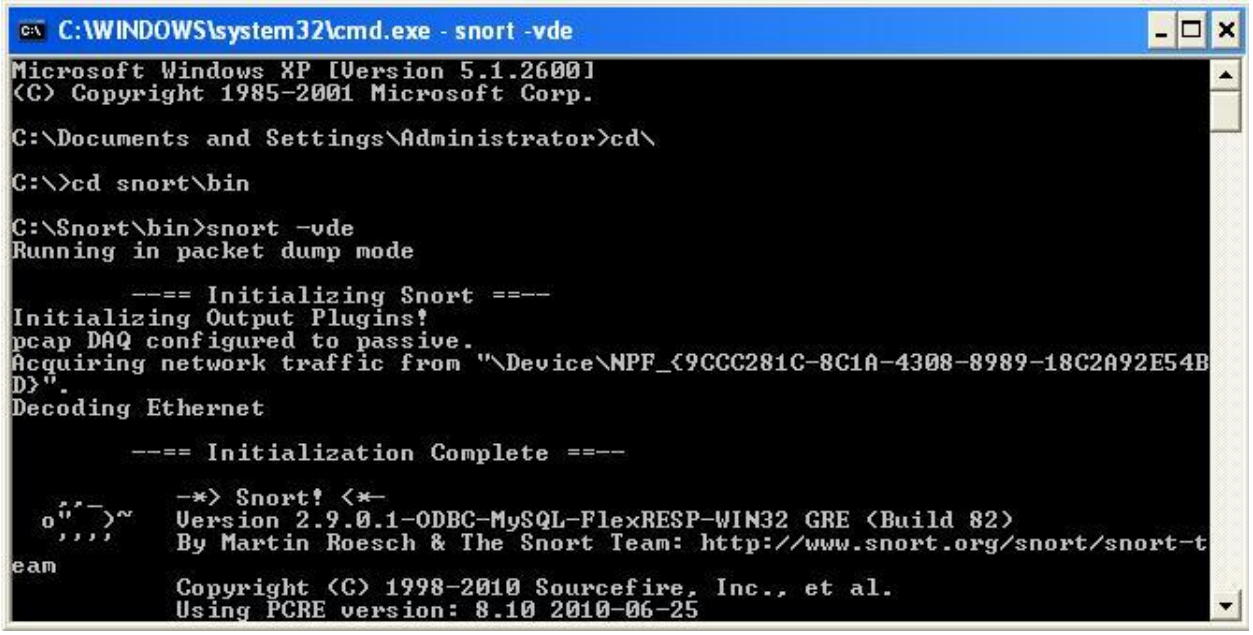
In application snort the run command line of sniffer mode will be in this form *c:\snort\bin\snort –v.* This instructs Snort to sniffer and shows TCP/IP packet headers to the screen.

c:\snort\bin\snort –vd. This instructs Snort to sniffer the packet data which is application data in transit as well as the TCP/IP packet headers and displays it on the snort screen.

While *c:\snort\bin\snort –vde* .This instructs Snort to sniffer and descriptive display includes headers, the packet data, and the data link layer headers.

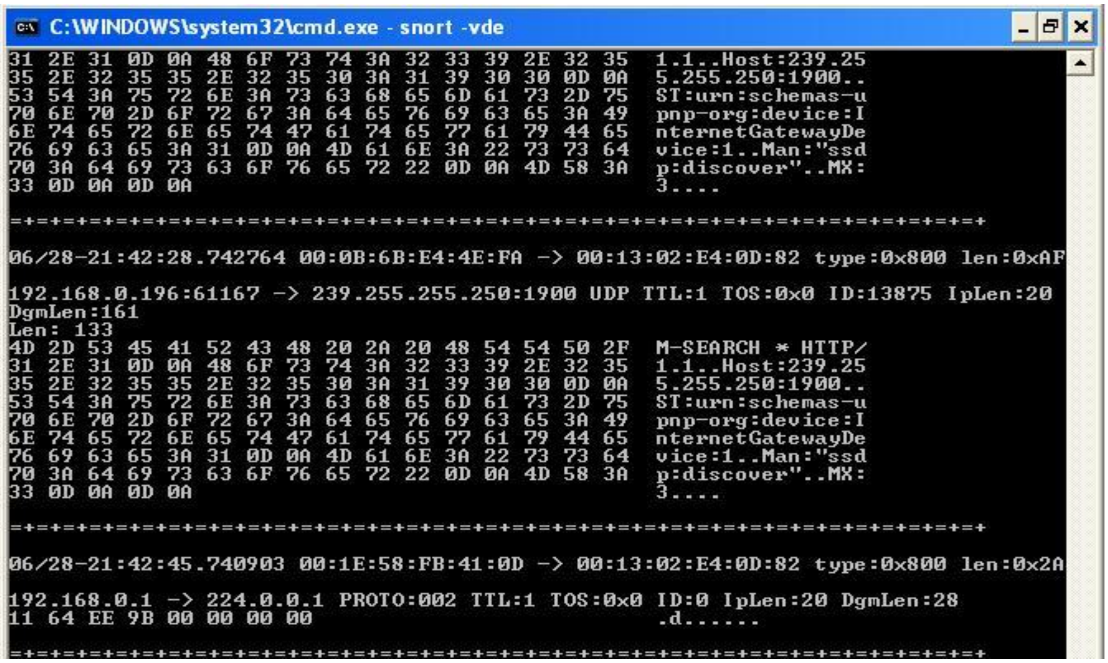
To do this, the request will be sent through high-level function a Libpcap library that contained several files such as the low-level packet capture, capture file reading, and capture file writing code of Winpcap. The Winpcap ran the appropriate library file to this current command line of the snort. After that was sent a request to low-level function the packet.dll to create and provide access for capturing a packet from network hardware directly and sending response of the capturing data through high level functions to the

request application to use it in a productive way according to the type of request rules of the snort in the proposed system. Figure 4.2 shows request for the command line of sniffer mode on the snort screen while Figure 4.3 shows response to the request by displaying capturing packet headers in snort screen.



```
C:\WINDOWS\system32\cmd.exe - snort -vde
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>cd\

C:\>cd snort\bin

C:\Snort\bin>snort -vde
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "\Device\NPF_{9CCC281C-8C1A-4308-8989-18C2A92E54B
D}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.0.1-ODBC-MySQL-FlexRESP-WIN32 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-t
eam
           Copyright (C) 1998-2010 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
```

Figure 4.2 the snort in sniffer mode



```
C:\WINDOWS\system32\cmd.exe - snort -vde
31 2E 31 0D 0A 48 6F 73 74 3A 32 33 39 2E 32 35  1.1..Host:239.25
35 2E 32 35 35 2E 32 35 30 3A 31 39 30 30 0D 0A  5.255.250:1900..
53 54 3A 75 72 6E 3A 73 63 68 65 6D 61 73 2D 75  ST:urn:schemas-u
70 6E 70 2D 6F 72 67 3A 64 65 76 69 63 65 3A 49  pnp-org:device:I
6E 74 65 72 6E 65 74 47 61 74 65 77 61 79 44 65  nternetGatewayDe
76 69 63 65 3A 31 0D 0A 4D 61 6E 3A 22 73 73 64  vice:1..Man:"ssd
70 3A 64 69 73 63 6F 76 65 72 22 0D 0A 4D 58 3A  p:discover"..MX:
33 0D 0A 0D 0A                                   3....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/28-21:42:28.742764 00:0B:6B:E4:4E:FA -> 00:13:02:E4:0D:82 type:0x800 len:0xAF

192.168.0.196:61167 -> 239.255.255.250:1900 UDP TTL:1 TOS:0x0 ID:13875 IpLen:20
DgmLen:161
Len: 133
4D 2D 53 45 41 52 43 48 20 2A 20 48 54 54 50 2F  M-SEARCH * HTTP/
31 2E 31 0D 0A 48 6F 73 74 3A 32 33 39 2E 32 35  1.1..Host:239.25
35 2E 32 35 35 2E 32 35 30 3A 31 39 30 30 0D 0A  5.255.250:1900..
53 54 3A 75 72 6E 3A 73 63 68 65 6D 61 73 2D 75  ST:urn:schemas-u
70 6E 70 2D 6F 72 67 3A 64 65 76 69 63 65 3A 49  pnp-org:device:I
6E 74 65 72 6E 65 74 47 61 74 65 77 61 79 44 65  nternetGatewayDe
76 69 63 65 3A 31 0D 0A 4D 61 6E 3A 22 73 73 64  vice:1..Man:"ssd
70 3A 64 69 73 63 6F 76 65 72 22 0D 0A 4D 58 3A  p:discover"..MX:
33 0D 0A 0D 0A                                   3....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/28-21:42:45.740903 00:1E:58:FB:41:0D -> 00:13:02:E4:0D:82 type:0x800 len:0x2A

192.168.0.1 -> 224.0.0.1 PROTO:002 TTL:1 TOS:0x0 ID:0 IpLen:20 DgmLen:28
11 64 EE 9B 00 00 00 00                          .d......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 4.3 the capturing packet headers in snort screen

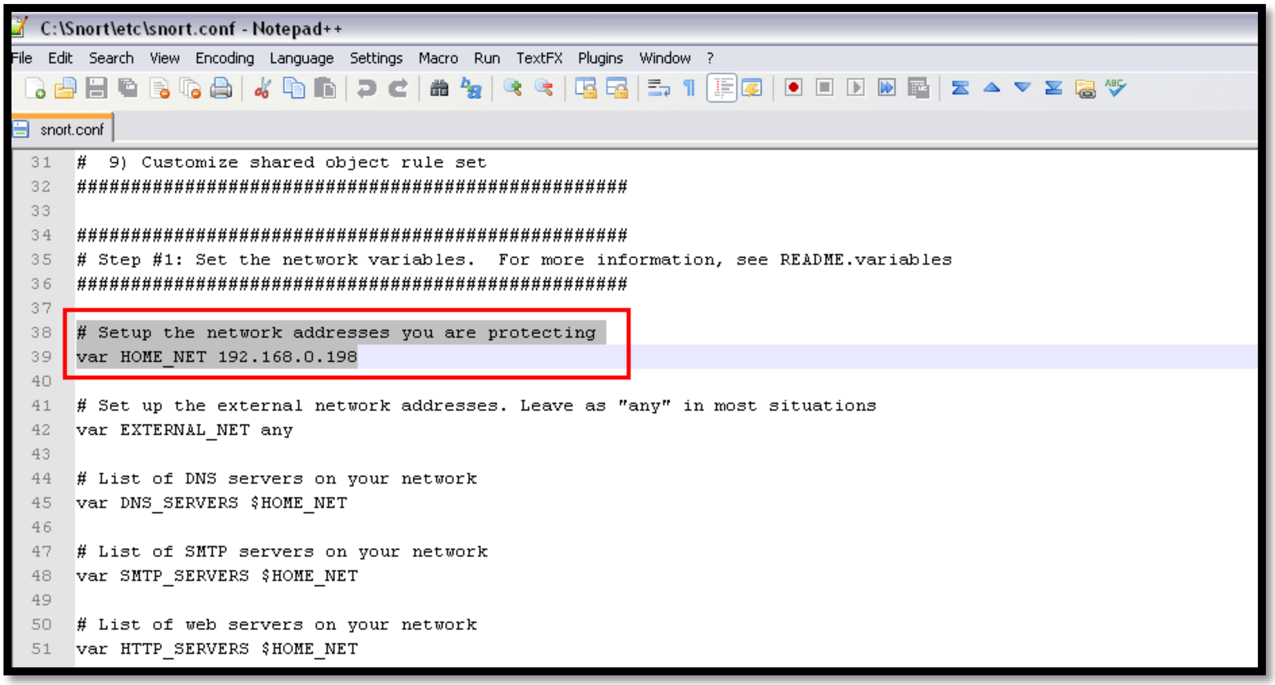
After the sniffer phase starts by request of the snort to Winpcap tool, now the NIPS phase will be started.

4.3 Implementation NIPS phase

To implement NIPS by using the snort tool in which an open source of a network intrusion prevention and detection system uses both a rule-based signature and anomaly inspection methods. After the snort tool is installed, then must modify the *"Snort.Conf"* file which is configuration files of the snort to suit requirements of the proposed system and the tools used with it. The *"Snort.Conf"* file is activated of selected rules file and this will apply the activation rules that be selected in the *"Snort.Conf"* file to each sniffer packet to decide if an action based upon the rule type in the file should be taken.

To run the snort in Network Intrusion prevention System (NIPS) mode and logger mode we have to follow the following instructions. This will do by configure Snort to run in its NIPS form, logging packets that trigger rules specified in the *"Snort.Conf"* file to disk. Snort runs in Logger mode, it collects every packet it sees and places it in a directory to the disk.

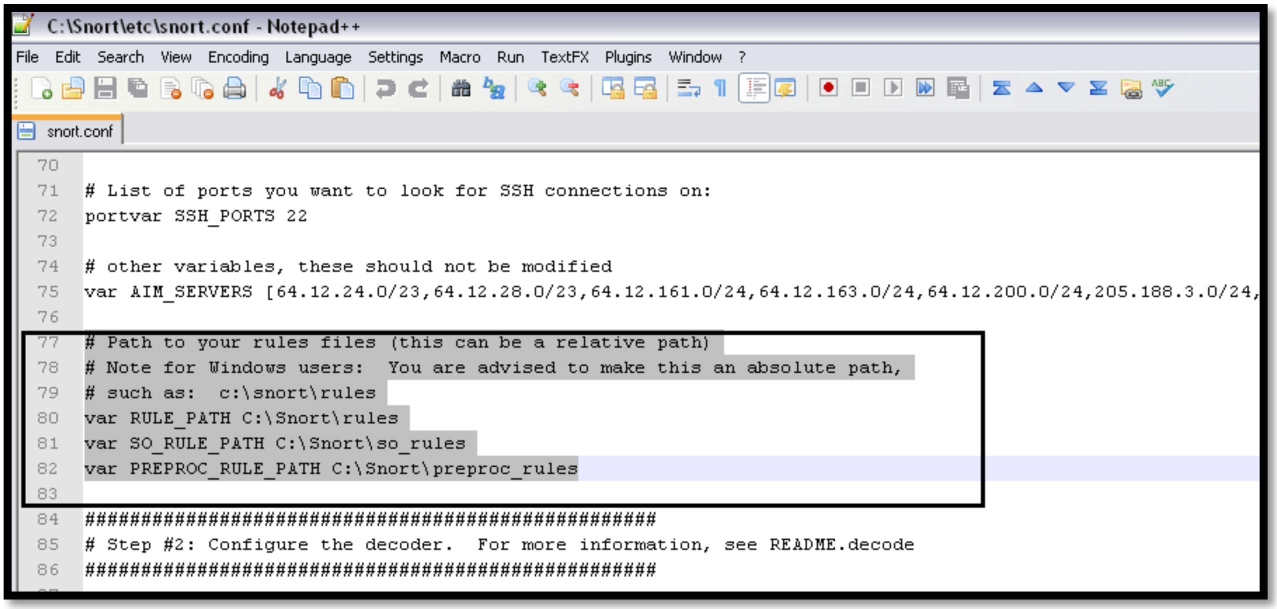
1. In the Snort configuration file , the network address that will be monitored has been set as shown in figure 4.4


C:\Snort\etc\snort.conf - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
snort.conf

```
#  9) Customize shared object rule set
####################################################

####################################################
# Step #1: Set the network variables.  For more information, see README.variables
####################################################

# Setup the network addresses you are protecting
var HOME_NET 192.168.0.198

# Set up the external network addresses. Leave as "any" in most situations
var EXTERNAL_NET any

# List of DNS servers on your network
var DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
var SMTP_SERVERS $HOME_NET

# List of web servers on your network
var HTTP_SERVERS $HOME_NET
```

Figure 4.4 The Config.File-Snort-Set the network address

2. The Rule Path in the configuration file of Snort has been also set. Please refer to figure 4.5


C:\Snort\etc\snort.conf - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
snort.conf

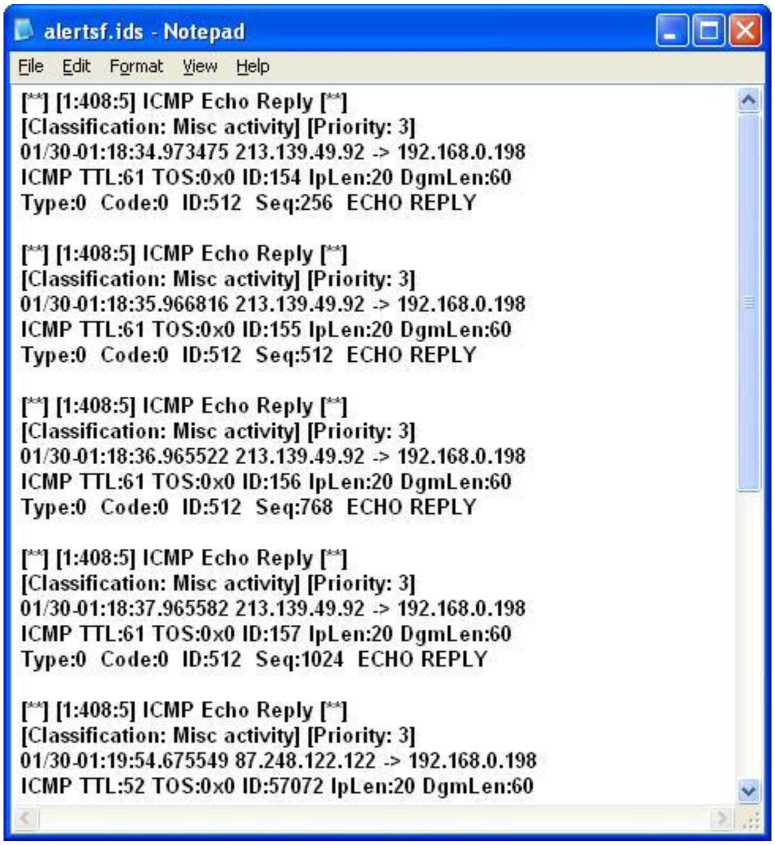
```

# List of ports you want to look for SSH connections on:
portvar SSH_PORTS 22

# other variables, these should not be modified
var AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,

# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\snort\rules
var RULE_PATH C:\Snort\rules
var SO_RULE_PATH C:\Snort\so_rules
var PREPROC_RULE_PATH C:\Snort\preproc_rules

####################################################
# Step #2: Configure the decoder.  For more information, see README.decode
####################################################
```

Figure 4.5 The Config.File-Snort-Set the Rule Path

Now by using instructions to enable Network Intrusion prevention System (NIPS) mode and logger mode is *c:\snort\bin\snort –vde -c c:\snort\etc\snort.conf -l c:\snort\log*. It will be saved as batch file to execute directly.

Where *-vde* mean sniffer and descriptive display includes headers, the packet data, and the data link layer headers as shown in Figure 4.3.
-c is use Rule file that identified in c:\snort\etc\snort.conf.
-l is log to directory which in the purpose system is the path c:\snort\log.

In the log file shown in Figure 4.6. It will save the alert in a simple format that will include a timestamp, alert message, source and destination IPs/ports. This detail of the log file will be saved in SQL database to be used in the next phase (advance analysis).



```
[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
01/30-01:18:34.973475 213.139.49.92 -> 192.168.0.198
ICMP TTL:61 TOS:0x0 ID:154 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:256  ECHO REPLY

[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
01/30-01:18:35.966816 213.139.49.92 -> 192.168.0.198
ICMP TTL:61 TOS:0x0 ID:155 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:512  ECHO REPLY

[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
01/30-01:18:36.965522 213.139.49.92 -> 192.168.0.198
ICMP TTL:61 TOS:0x0 ID:156 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:768  ECHO REPLY

[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
01/30-01:18:37.965582 213.139.49.92 -> 192.168.0.198
ICMP TTL:61 TOS:0x0 ID:157 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512  Seq:1024  ECHO REPLY

[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
01/30-01:19:54.675549 87.248.122.122 -> 192.168.0.198
ICMP TTL:52 TOS:0x0 ID:57072 IpLen:20 DgmLen:60
```

Figure 4.6 log file of the snort

4.4 Implementation of an advanced analysis phase

As it has been mentioned in the previous chapter, despite the power of the Snort as the powerful tool in the field of intrusion prevention systems, it needs to support and more accurate to detect internal network attack and improve the anomaly detection approach.

In this phase we need deep analysis for log file of the snort so we can find hidden behavior to the connection traffic between the source and destination.

The reliance on the ability of the human in the analysis of this file is not an effective way in addition it lacks the accuracy. This is because there are many records in this file and the size of database is huge and this requires a long time for analysis as well as it needs a high skill of an analyst to discover all the hidden behavior.

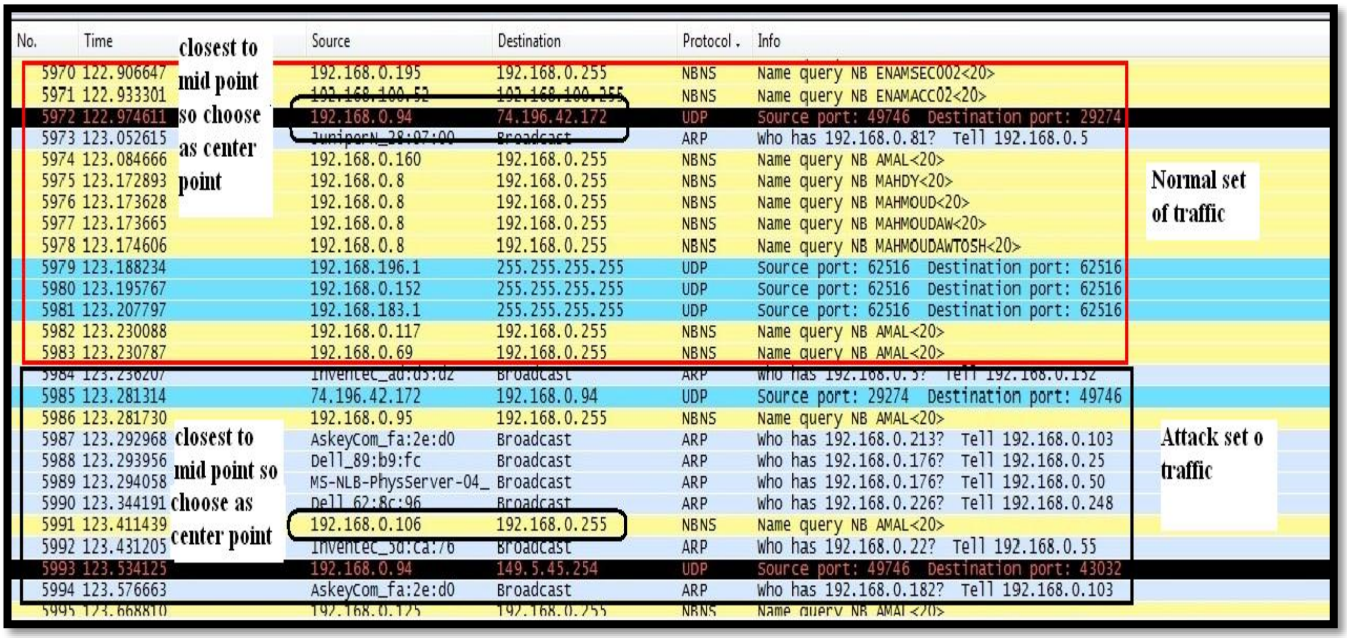
As the suggestion is to use data mining algorithms due to its effectiveness at work with a huge database, helps an analyst to discover new rules from a hidden behavior of traffic between the source and destination that snort tool can't see it as obvious rules. To do this, we will depend on two sub-phases of data mining approach. The first sub-phase is improved K-mean algorithm to classify the log file of snort to clusters in an effective manner. From the resultant clusters, records will be taken as an input to the second sub-phase which is the FP-Growth Algorithm that will generate the frequent patterns to these records and that helps to create new rules from these patterns which therefore leads to improve the snort tool. Please refer to the steps bellow.

This phase is represented by the following steps:

1. As mentioned in chapter three, the first step of execution, is to calculate manually the initial center points, by plotting candidates points and calculating the midpoint from two correspond points of the boundary cluster. Then choose the point closest to the midpoint and considered

2. as the center point to this cluster. The initial number of clusters K=2, as shown bellow in figure 4.7 where there is two group (clusters), the normal and the attack.

   The classification of the two groups (normal or attack) depends on the analytical tools that show valid and block IP`s according to certain rules.

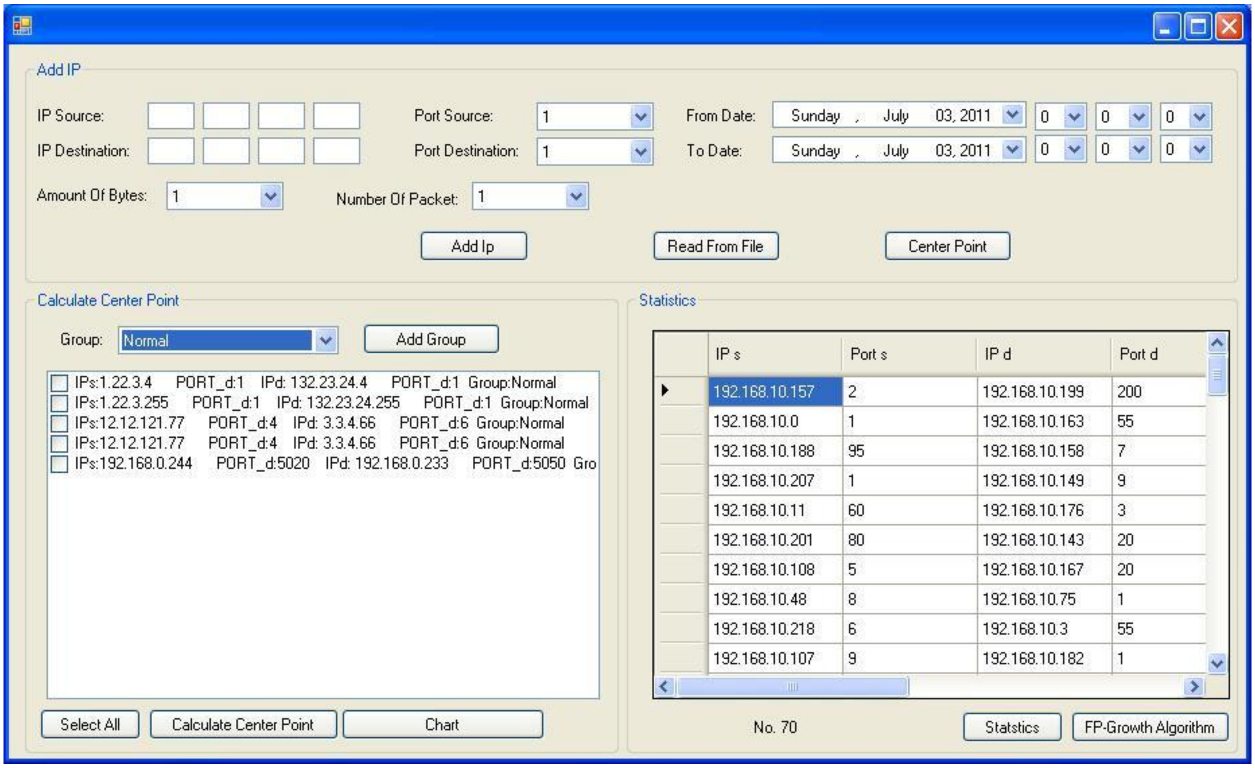

closest to mid point so choose as center point
Normal set of traffic
closest to mid point so choose as center point
Attack set o traffic

No.	Time	Source	Destination	Protocol	Info
5970	122.906647	192.168.0.195	192.168.0.255	NBNS	Name query NB ENAMSEC002<20>
5971	122.933301	192.168.100.52	192.168.100.255	NBNS	Name query NB ENAMACC02<20>
5972	122.974611	192.168.0.94	74.196.42.172	UDP	Source port: 49746 Destination port: 29274
5973	123.052615	JuniperN_28:07:00	Broadcast	ARP	Who has 192.168.0.81? Tell 192.168.0.5
5974	123.084666	192.168.0.160	192.168.0.255	NBNS	Name query NB AMAL<20>
5975	123.172893	192.168.0.8	192.168.0.255	NBNS	Name query NB MAHDY<20>
5976	123.173628	192.168.0.8	192.168.0.255	NBNS	Name query NB MAHMOUD<20>
5977	123.173665	192.168.0.8	192.168.0.255	NBNS	Name query NB MAHMOUDAW<20>
5978	123.174606	192.168.0.8	192.168.0.255	NBNS	Name query NB MAHMOUDAWTOSH<20>
5979	123.188234	192.168.196.1	255.255.255.255	UDP	Source port: 62516 Destination port: 62516
5980	123.195767	192.168.0.152	255.255.255.255	UDP	Source port: 62516 Destination port: 62516
5981	123.207797	192.168.183.1	255.255.255.255	UDP	Source port: 62516 Destination port: 62516
5982	123.230088	192.168.0.117	192.168.0.255	NBNS	Name query NB AMAL<20>
5983	123.230787	192.168.0.69	192.168.0.255	NBNS	Name query NB AMAL<20>
5984	123.236207	Inventec_ad:d5:d2	Broadcast	ARP	Who has 192.168.0.5? Tell 192.168.0.152
5985	123.281314	74.196.42.172	192.168.0.94	UDP	Source port: 29274 Destination port: 49746
5986	123.281730	192.168.0.95	192.168.0.255	NBNS	Name query NB AMAL<20>
5987	123.292968	AskeyCom_fa:2e:d0	Broadcast	ARP	Who has 192.168.0.213? Tell 192.168.0.103
5988	123.293956	Dell_89:b9:fc	Broadcast	ARP	Who has 192.168.0.176? Tell 192.168.0.25
5989	123.294058	MS-NLB-PhysServer-04_	Broadcast	ARP	Who has 192.168.0.176? Tell 192.168.0.50
5990	123.344191	Dell_62:8c:96	Broadcast	ARP	Who has 192.168.0.226? Tell 192.168.0.248
5991	123.411439	192.168.0.106	192.168.0.255	NBNS	Name query NB AMAL<20>
5992	123.431205	Inventec_5d:ca:76	Broadcast	ARP	Who has 192.168.0.22? Tell 192.168.0.55
5993	123.534125	192.168.0.94	149.5.45.254	UDP	Source port: 49746 Destination port: 43032
5994	123.576663	AskeyCom_fa:2e:d0	Broadcast	ARP	Who has 192.168.0.182? Tell 192.168.0.103
5995	123.668810	192.168.0.125	192.168.0.255	NBNS	Name query NB AMAL<20>

Figure 4.7 Training data to choose center point

3. The second step is to enter or browse the snort log file by clicking on the *“Read From File”* button in order to read the data and used it in the first sub-phase of data mining algorithm as shown in the Figure 4.8.

   Note: It is also possible to enter/set the field’s values in the bellow screen manually.



IP Source:
IP Destination:
Port Source: 1
Port Destination: 1
From Date: Sunday , July 03, 2011 0 0 0
To Date: Sunday , July 03, 2011 0 0 0
Amount Of Bytes: 1
Number Of Packet: 1

Group: Normal

- □ IPs:1.22.3.4 PORT_d:1 IPd: 132.23.24.4 PORT_d:1 Group:Normal
- □ IPs:1.22.3.255 PORT_d:1 IPd: 132.23.24.255 PORT_d:1 Group:Normal
- □ IPs:12.12.121.77 PORT_d:4 IPd: 3.3.4.66 PORT_d:6 Group:Normal
- □ IPs:12.12.121.77 PORT_d:4 IPd: 3.3.4.66 PORT_d:6 Group:Normal
- □ IPs:192.168.0.244 PORT_d:5020 IPd: 192.168.0.233 PORT_d:5050 Gro

	IP s	Port s	IP d	Port d
▶	192.168.10.157	2	192.168.10.199	200
	192.168.10.0	1	192.168.10.163	55
	192.168.10.188	95	192.168.10.158	7
	192.168.10.207	1	192.168.10.149	9
	192.168.10.11	60	192.168.10.176	3
	192.168.10.201	80	192.168.10.143	20
	192.168.10.108	5	192.168.10.167	20
	192.168.10.48	8	192.168.10.75	1
	192.168.10.218	6	192.168.10.3	55
	192.168.10.107	9	192.168.10.182	1

No. 70

Figure 4.8 the interface of the proposed system

4. The first sub-phase of data mining is the improved K-mean algorithm as was mentioned in chapter three. It receives input data and distributes it into the clusters. An example of the input data is shown bellow, where the input data are: IP source and port, From Datetime, IP destination and port, and To Datetime.

```
192.168.10.218,15,6/12/2011 3:22:17 PM,192.168.10.3,5,6/12/2011 3:22:43 PM

192.168.10.127,11,6/12/2011 3:22:17 PM,192.168.10.167,2,6/12/2011 3:22:39 PM

192.168.10.72,17,6/12/2011 3:22:17 PM,192.168.10.23,17,6/12/2011 3:22:29 PM

192.168.10.25,8,6/12/2011 3:22:17 PM,192.168.10.202,14,6/12/2011 3:22:35 PM

                                    :

                                    :

                                    :

192.168.10.25,3,6/12/2011 3:22:17 PM,192.168.10.170,1,6/12/2011 3:22:29 PM
```

5. Previous definition in chapter 3, it has been specified where it is the center point which will be used to calculate the distance between the center point for each cluster and the input points. In case of points not belong to any of the defined clusters; a window will appear to request from the analyst (End User) to take the appropriate decision; either creating a new group and its first point will be a center point, or choosing the closest clusters in case of having point intermediate between clusters. Thus; the new created clusters will be added to the previous clusters which will be used by the analyst in making future decisions for the input of new data.

6. In the proposed system, there is the possibility of recalculate the center point for each cluster after the distribution of the new input data by clicking on the “*Calculate Center Point” button* in the interface of the proposed system.

   In addition there is an option to click on the *"chart"* button in order to be able to plot these points with its center point for each selected cluster. As shown in the Figure 4.9 which displays plotter to some selected records of a probe cluster. Plotter's points are represented by pair (IPsPORTs, IPdPORTd), where IPsPORTs is a last digit of IP source joined with a digit of port source while IPdPORTd is a last digit of IP destination joined with a digit of port destination. For example the point (113,421) ,the value 113 belong to source IP:192.168.0.11 and port:3 while the value 421 belong to destination IP: 192.168.0.42 port:1 .

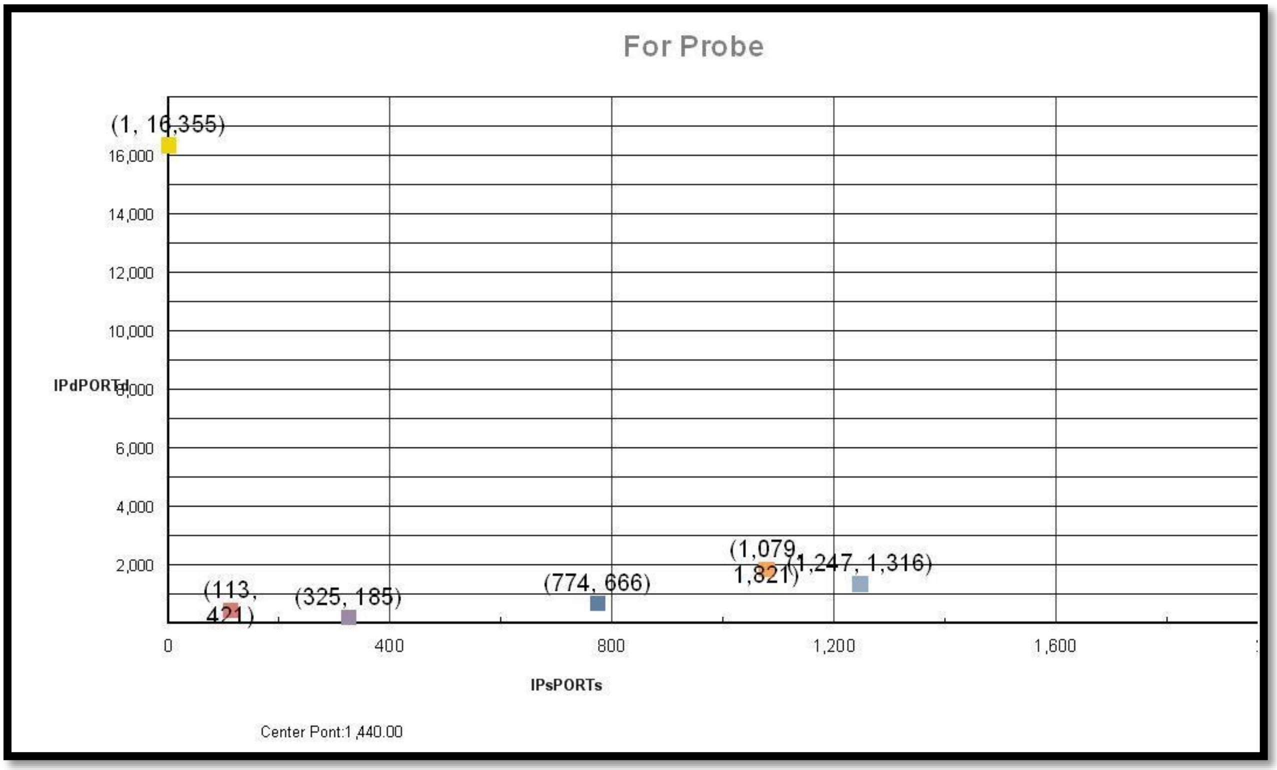
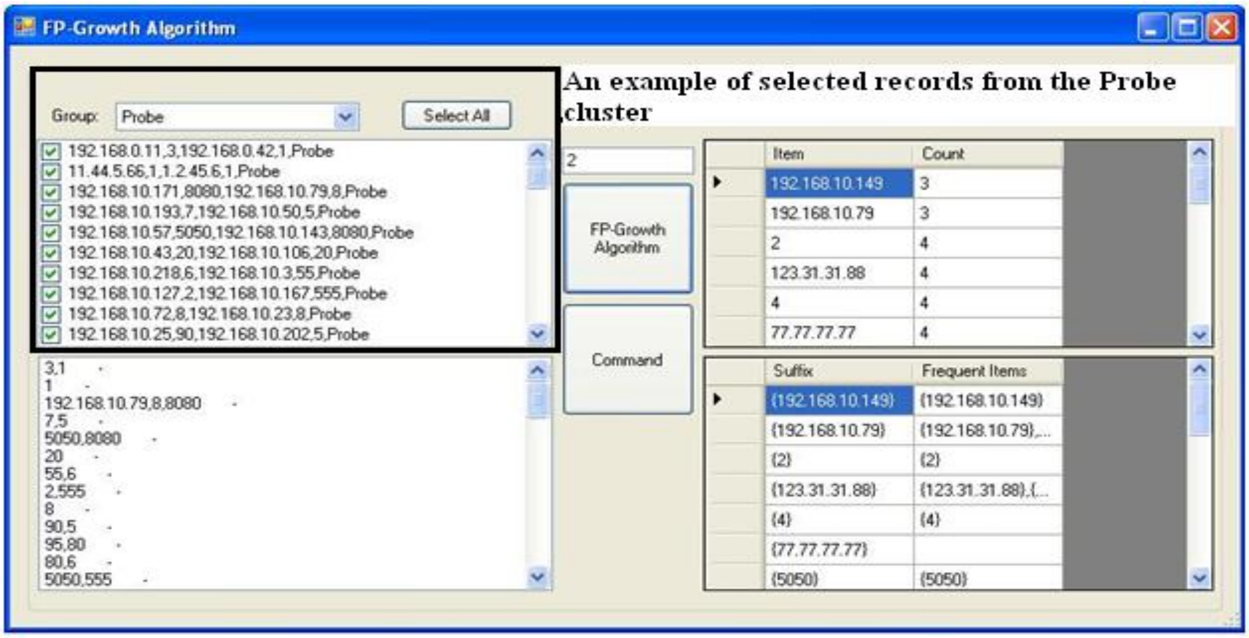


Figure 4.9 the chart window of   Probe cluster

7. After the distribution of the data on their clusters and create the new cluster in case it’s requiring that, in addition to recalculate the center point for each cluster. Then selected records from one of these specific clusters as shown in figure 4.10 so it can be used in the creation of frequent pattern by using the second sub-phase of the data mining algorithm.



Group: Probe

☑ 192.168.0.11,3,192.168.0.42,1,Probe
☑ 11.44.5.66,1,1.2.45.6,1,Probe
☑ 192.168.10.171,8080,192.168.10.79,8,Probe
☑ 192.168.10.193,7,192.168.10.50,5,Probe
☑ 192.168.10.57,5050,192.168.10.143,8080,Probe
☑ 192.168.10.43,20,192.168.10.106,20,Probe
☑ 192.168.10.218,6,192.168.10.3,55,Probe
☑ 192.168.10.127,2,192.168.10.167,555,Probe
☑ 192.168.10.72,8,192.168.10.23,8,Probe
☑ 192.168.10.25,90,192.168.10.202,5,Probe

```
3,1
1
192.168.10.79,8,8080
7,5
5050,8080
20
55,6
2,555
8
90,5
95,80
80,6
5050,555
```

An example of selected records from the Probe cluster

2

Item	Count
192.168.10.149	3
192.168.10.79	3
2	4
123.31.31.88	4
4	4
77.77.77.77	4

Suffix	Frequent Items
{192.168.10.149}	{192.168.10.149}
{192.168.10.79}	{192.168.10.79},...
{2}	{2}
{123.31.31.88}	{123.31.31.88},{...
{4}	{4}
{77.77.77.77}	
{5050}	{5050}

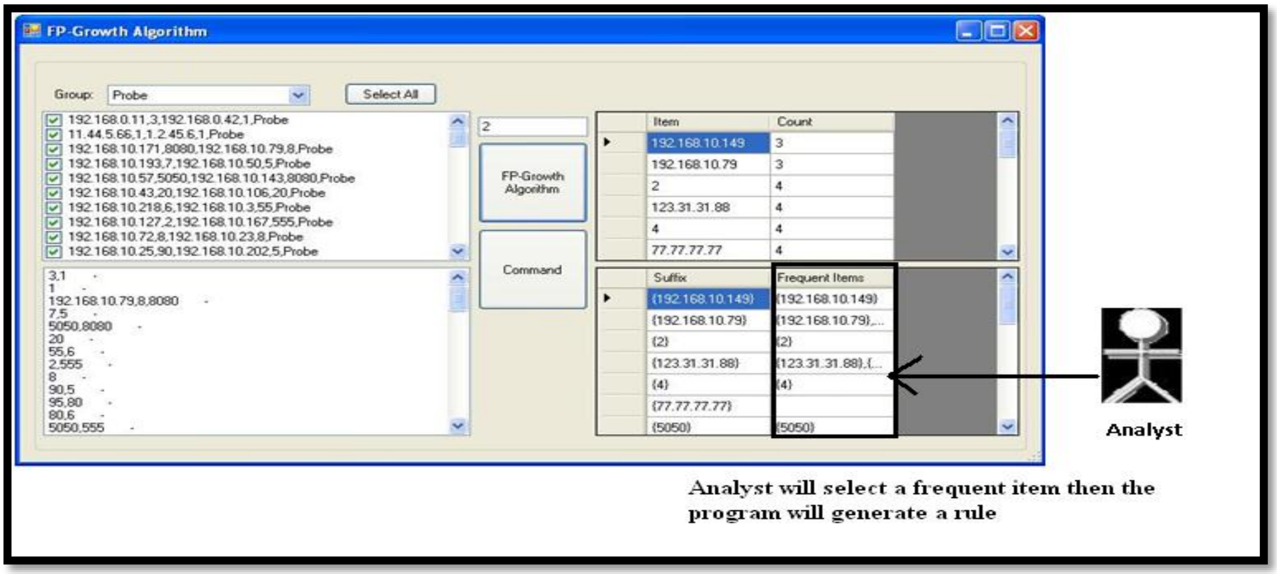
Figure 4.10 Example of selected records from specific cluster.

8. The second sub-phase of the data mining algorithm is FP-growth algorithm as was mentioned in the previous chapter. The aim of this step is to build the tree from the itemset of selected records of the specified cluster. These selected records contain IP source, port source, IP destination, port destination, and name of the cluster belongs to the selected records. Then it is extracted sub tree for each item set in the main tree. From each of these subtrees, a group of the frequent pattern *"it is titled frequent items in this system"* and suffix, which is a main item that linked to its sub-tree can be created.

9. This will be done by click on *"FP-Growth Algorithm"* button in the interface of the proposed system.

10. These frequent patterns that were generated belonged to the selected cluster and from them it is possible to create new rules. Now the role

11. of the analyst comes to generate a rule from the selected frequent items. The selected frequent items must be characterized by its contents IP source, port source, IP destination, and port destination as shown in the Figure 4.11.



☑ 192.168.0.11,3,192.168.0.42,1,Probe
☑ 11.44.5.66,1,1.2.45.6,1,Probe
☑ 192.168.10.171,8080,192.168.10.79,8,Probe
☑ 192.168.10.193,7,192.168.10.50,5,Probe
☑ 192.168.10.57,5050,192.168.10.143,8080,Probe
☑ 192.168.10.43,20,192.168.10.106,20,Probe
☑ 192.168.10.218,6,192.168.10.3,55,Probe
☑ 192.168.10.127,2,192.168.10.167,555,Probe
☑ 192.168.10.72,8,192.168.10.23,8,Probe
☑ 192.168.10.25,90,192.168.10.202,5,Probe

```
3,1        -
1        -
192.168.10.79,8,8080        -
7,5        -
5050,8080        -
20        -
55,6        -
2,555        -
8        -
90,5        -
95,80        -
80,6        -
5050,555        -
```

2

	Item	Count
▶	192.168.10.149	3
	192.168.10.79	3
	2	4
	123.31.31.88	4
	4	4
	77.77.77.77	4

	Suffix	Frequent Items
▶	{192.168.10.149}	{192.168.10.149}
	{192.168.10.79}	{192.168.10.79},...
	{2}	{2}
	{123.31.31.88}	{123.31.31.88},{...
	{4}	{4}
	{77.77.77.77}	
	{5050}	{5050}

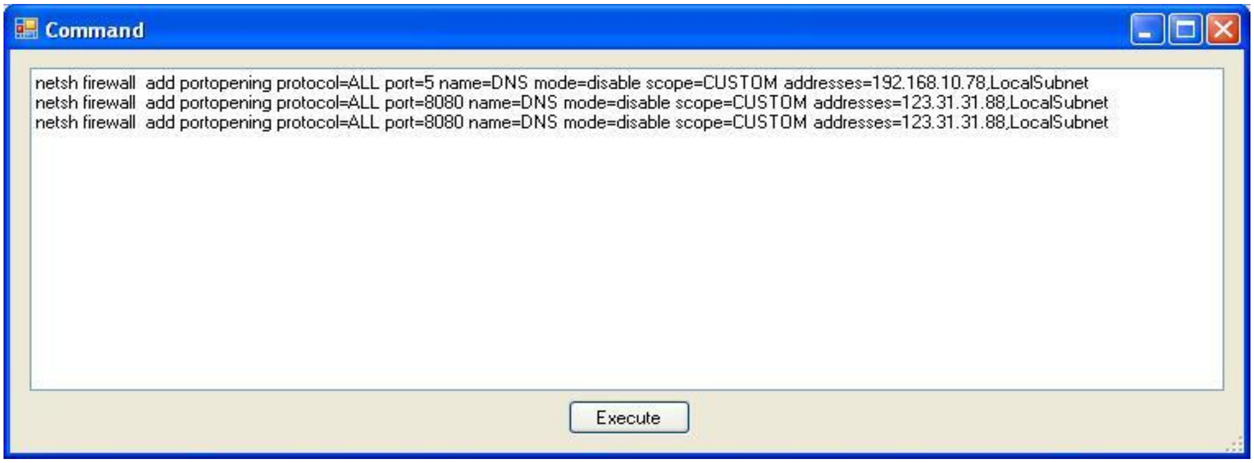
Analyst will select a frequent item then the program will generate a rule

Figure 4.11 window of FP-growth in the proposed system

12. Generated new rules will be by clicking on *"Command"* button in the FP-growth window. This will show command window with command lines that could be used to change configuration of firewall. This command is will prevent/block the IP source that targeted port

13. destination from accessing the specified IP destination on this port depending on the selected frequent items.

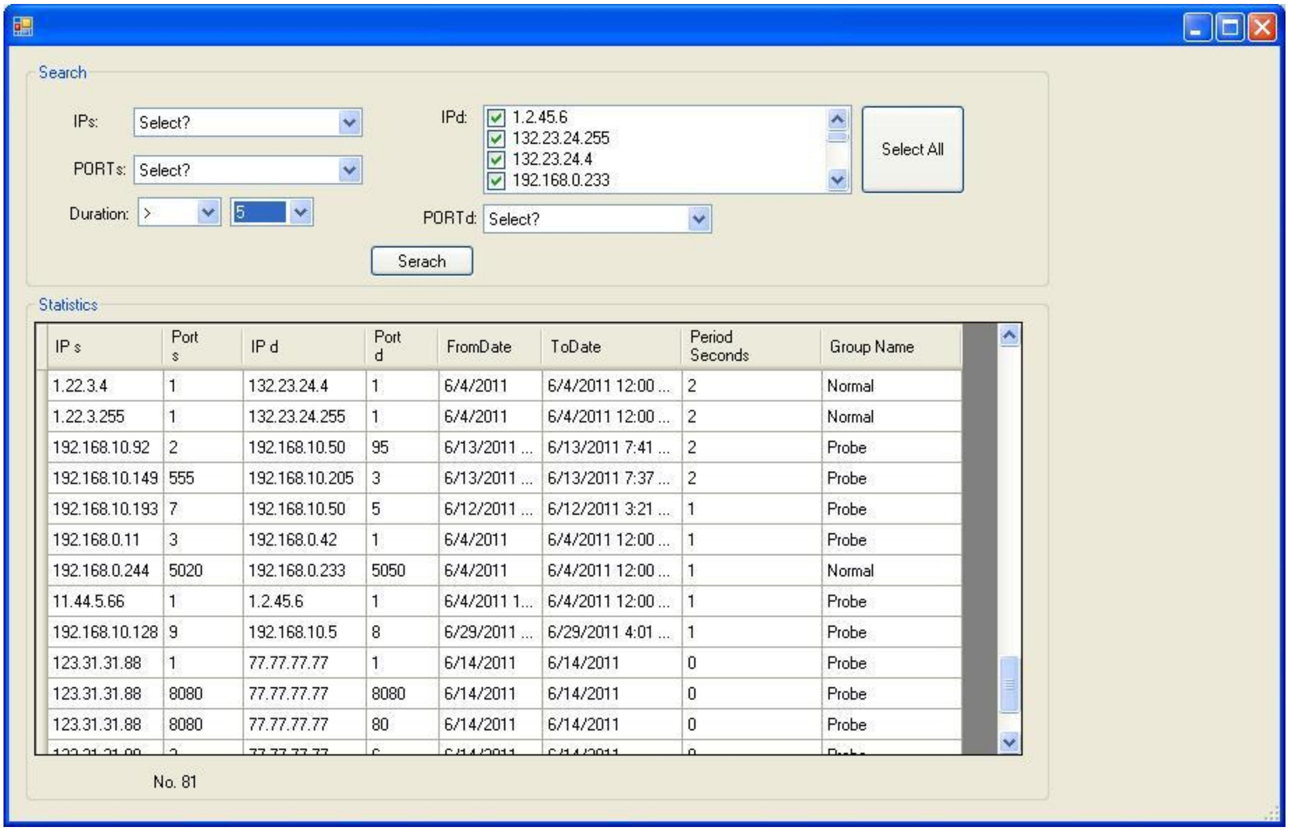
14. When click on *"Execute"* button in the command window, will do this prevention to the IP source from access by this IP that belongs to this port destination. As shown in the Figure 4.12.



```
netsh firewall  add portopening protocol=ALL port=5 name=DNS mode=disable scope=CUSTOM addresses=192.168.10.78,LocalSubnet
netsh firewall  add portopening protocol=ALL port=8080 name=DNS mode=disable scope=CUSTOM addresses=123.31.31.88,LocalSubnet
netsh firewall  add portopening protocol=ALL port=8080 name=DNS mode=disable scope=CUSTOM addresses=123.31.31.88,LocalSubnet
```

Figure 4.12 Execute rules in the proposed system

15. One of the additional features/options in the proposed system is shown in Figure 4.13. It is considered as advanced statistics screen that can help the analyst to take the appropriate decisions. The analyst can search which IP source targets the IP destination by any port, as well as which IP is used to get out from any port, which IP is used to enter through any port in addition to the duration of time which is represented by seconds. The analyst role here is to only fill the data (Criteria) and click the *“Search”* button. The resultant data will then be displayed in a statistical table.



IPs: Select?

PORTs: Select?

Duration: > 5

IPd: ☑ 1.2.45.6 ☑ 132.23.24.255 ☑ 132.23.24.4 ☑ 192.168.0.233

PORTd: Select?

IP s	Port s	IP d	Port d	FromDate	ToDate	Period Seconds	Group Name
1.22.3.4	1	132.23.24.4	1	6/4/2011	6/4/2011 12:00 ...	2	Normal
1.22.3.255	1	132.23.24.255	1	6/4/2011	6/4/2011 12:00 ...	2	Normal
192.168.10.92	2	192.168.10.50	95	6/13/2011 ...	6/13/2011 7:41 ...	2	Probe
192.168.10.149	555	192.168.10.205	3	6/13/2011 ...	6/13/2011 7:37 ...	2	Probe
192.168.10.193	7	192.168.10.50	5	6/12/2011 ...	6/12/2011 3:21 ...	1	Probe
192.168.0.11	3	192.168.0.42	1	6/4/2011	6/4/2011 12:00 ...	1	Probe
192.168.0.244	5020	192.168.0.233	5050	6/4/2011	6/4/2011 12:00 ...	1	Normal
11.44.5.66	1	1.2.45.6	1	6/4/2011 1...	6/4/2011 12:00 ...	1	Probe
192.168.10.128	9	192.168.10.5	8	6/29/2011 ...	6/29/2011 4:01 ...	1	Probe
123.31.31.88	1	77.77.77.77	1	6/14/2011	6/14/2011	0	Probe
123.31.31.88	8080	77.77.77.77	8080	6/14/2011	6/14/2011	0	Probe
123.31.31.88	8080	77.77.77.77	80	6/14/2011	6/14/2011	0	Probe

No. 81

Figure 4.13 the window of advanced statistics in the proposed system

4.5 The testing

The system is a LAN network that consists of 4 nodes connected to one main server. The snort tool was installed on the server in order to monitor the in and out traffic comes through the workstations. Below are the steps of testing and executing the system:

1. Execute the snort tool as an inline mode (NIPS) that sniffer network traffic and take the appropriate decisions pass or drop the traffic depending on the snort rules and snort alerts which are saved in the Snort log file.

2. Then begin to call or enter the Snort log file to be analyzed by the proposed system.

The first and second steps are illustrated in the two cases below:

The first case: There are 10 records were taken from the first log file that was extracted from the execution of the snort file on 15/6/2011 as shown below.

```
1. 06/15-15:54:48.206470  [**] [1:3691:2] CHAT Yahoo Messenger Message [**] [Classification: pass normal traffic] [Priority: 1] {TCP} 192.168.0.198:1081 -> 67.195.186.244:5050

2. 06/15-19:42:21.484102  [**] [1:12286:5] WEB-CLIENT PCRE character class double free overflow attempt [**] [Classification: Attempted User Privilege Gain] [Priority: 1] {TCP} 209.85.146.95:80 -> 192.168.0.198:3676

3. 06/15-21:10:36.261720  [**] [1:254:8] DNS SPOOF query response with TTL of 1 min. and no authority [**] [Classification: Bad Traffic] [Priority: 2] {UDP} 98.136.154.148:53 -> 192.168.0.198:64654

4. 06/15-21:12:11.609882  [**] [1:3691:2] CHAT Yahoo Messenger Message [**] [Classification: pass normal traffic] [Priority: 1] {TCP} 192.168.0.198:4096 -> 67.195.187.232:5050

5. 06/15-21:12:16.483448  [**] [1:3691:2] CHAT Yahoo Messenger Message [**] [Classification: pass normal traffic] [Priority: 1] {TCP} 192.168.0.198:4096 -> 67.195.187.232:5050
```

6. 06/15-21:12:22.431683 [**] [1:3691:2] CHAT Yahoo Messenger Message [**] [Classification: pass normal traffic] [Priority: 1] {TCP} 192.168.0.198:4096 -> 67.195.187.232:5050
7. 06/15-21:12:33.907289 [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 98.136.154.147:80 -> 192.168.0.198:4727
8. 06/15-21:12:34.770087 [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 76.13.222.36:80 -> 192.168.0.198:4731
9. 06/15-21:12:35.553947 [**] [1:1394:12] SHELLCODE x86 inc ecx NOOP [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 98.136.154.147:80 -> 192.168.0.198:4727
10. 06/15-00:07:08.156157 [**] [1:3692:2] CHAT Yahoo Messenger File Transfer Initiation Request [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {TCP} 192.168.0.198: 27374 -> 98.136.112.30:80

Here the snort tool makes a decision to drop the third traffic, classified it as *"bad traffic"* and displayed it either as an alert shown in a screen or as log file or both.

All the remaining records were set to pass. However the system showed an alert for the second and tenth records as it was an attempt to use the privileged gain.

The overview to decisions snort about these transactions is passing the traffic that in general is normal or drop the bad traffic depend on snort rules. As a result there are nine traffics classified as a normal case while one traffic is dropped and classified as an attack case depending on the snort tool that analyzes content of this traffic and used ports.

Now by the proposed system, it will apply this log file to advanced analysis, which two sub-phases of data mining approaches. In first sub-phase which is improved K-mean algorithm. The input records were classified as follows:

- The Records 1, 4, 5, and 6 put in normal cluster based on the initial center point that was chosen for normal traffic. In this case K=2.
- The Records 3 put in attack cluster based on second initial center point that was chosen for attack traffic. In this case K=2.
- The Records 7, 8, 9 put in the test cluster because it was out of the normal or attack groups, so the Analyst created based on that a new cluster called Test and chose record 7 as an initial center point. In this case K will become equal to 3.
- The Records 2, 10 were considered as intermediate records between normal and attack cluster since it was an attempt to use the gain privileges, so based on that the analyst then made a decision to put these two points in new group that will named Probe cluster and chose record 2 as an initial center points. In this case K will become equal to 4.

The second step is to use the FP-growth algorithm in order to selected records from the four clusters mentioned above. Analyst will select records from the attack, probe, and test clusters. Then execute the FP-growth algorithm which will finally display results in table that has suffix and frequent items.

Examples of frequent items are shown below:

IPs PORTs IPd PORTd

“98.136.154.147, 80, 192.168.0.198, 4727” frequent item appears more than twice

“98.136.154.147, 27374, 192.168.0.198, 4727” frequent item appears once

“98.136.154.148, 3676, 192.168.0.198, 4727” frequent item appears once

The patterns above indicate that the IP source 98.136.154.147 that use port 80 to target IP 192.168.0.198 through port 4727 also IP source 98.136.154.147 uses other port 27374 to access to the same IP 192.168.0.198, 4727 and port destination. The other IP source 98.136.154.148 by port 3676 targets the same IP 192.168.0.198.

Because the IP source targeted the same IP destination but with two different ports one of them is legal which is port 80 and the other is illegal because its signature is for the SubSeven worm which is port 27374. Therefore as a result of that the system prevented the IP source from accessing the IP destination by using a command line firewall.

The final results became as:

- 4 records were set in normal cluster.
- 1 record was set in attack cluster.
- 3 records were set in test cluster.
- 2 records were set in probe cluster.

Eventually, by using these two sub-phases of data mining make it possible to discover rules that the snort was not able to detect them.

The second case: There are several records that were taken from the other log file resulted from executing the snort in a different date. It was generated by using the ping Dos command line utility that can help to determine whether or not a particular network resource is responding on a network. This kind of test was done on IP 192.168.0.198 and 192.168.0.195 targeting the destination IP 192.168.0.196 on a different date but approximately the same time as shown below.

```
1. 01/26-21:03:53.133591  [**] [1:408:5] ICMP Echo Reply [**] [Classification:
Misc activity] [Priority: 3] {ICMP} 192.168.0.196-> 192.168.0.198

2. 01/26-21:03:54.112216  [**] [1:408:5] ICMP Echo Reply [**] [Classification:
Misc activity] [Priority: 3] {ICMP} } 192.168.0.196-> 192.168.0.198

3. 01/27-21:03:55.136140  [**] [1:408:5] ICMP Echo Reply [**] [Classification:
Misc activity] [Priority: 3] {ICMP} } 192.168.0.196 -> 192.168.0.198
```

4. 01/27-21:03:56.140249 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} } 192.168.0.196-> 192.168.0.198
5. 01/27-21:04:34.630793 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.196-> 192.168.0.198
6. 01/28-21:04:35.631639 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} } 192.168.0.196-> 192.168.0.198
7. 01/28-21:04:36.635904 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} } 192.168.0.196-> 192.168.0.195
8. 01/28-21:04:37.637671 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} } 192.168.0.196-> 192.168.0.198
9. 01/29-21:05:23.297750 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} } 192.168.0.196-> 192.168.0.195
10. 01/29-21:06:24.300824 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} } 192.168.0.196-> 192.168.0.195
11. 01/29-21:06:25.304843 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.196-> 192.168.0.195

```
12.  01/29-21:07:26.307814   [**]  [1:408:5]  ICMP  Echo  Reply  [**]
[Classification:  Misc  activity]  [Priority:  3]  {ICMP}  192.168.0.196->
192.168.0.198
```

Here the snort tool logs in file that IP destination 192.168.0.196 reply of echo ping request of IP source 192.168.0.198, 192.168.0.195. However, there is no alert by the snort tool about these frequent testing.

The proposed system will use this log file in the two sub-phases of data mining approach for performing advanced analysis. In first sub-phase which is improved K-mean algorithm. The input records classified most of the records and put them in probe cluster except the records that have IP 192.168.0.195 was put in test cluster that is defined previously by analyst in the first case.

Now by using second sub-phase FP-growth algorithm selected recently records of a probe and a test cluster. The execution FP-growth algorithm will appear as final result table frequent item. As a result these frequent items will be as following:

IPs   IPd

“192.168.0.195, 192.168.0.198” frequent item appears Three times in this test

“192.168.0.198, 192.168.0.196” frequent item appears Six times

“192.168.0.196, 192.168.0.195” frequent item appears Three times

“192.168.0.195” frequent item appears Three times

“192.168.0.198” frequent item appears Ten times

“192.168.0.196” frequent item appears Seven times

The patterns above indicate that the IP source 192.168.0.195 related with IP source 192.168.0.198, because the IP source 168.0.198 and IP source 192.168.0.195 access the same destination IP which is 192.168.0.196. As a result the data mining can discover these frequent tests and show a hidden relation between “192.168.0.195, 192.168.0.198” which may be came from same source of the malicious user and target IP destination 192.168.0.196.

Chapter Five

Conclusion and Future Work

5.1. Introduction

Our overall objective is to develop NIPS into a general framework for the defending against attacks and threats inside and outside the computer network system. The primary goal is to discover the process of suspicious probing inside the network before it launches any network attacks with devastating results.

The most network policies are not very explicit in defining rules and codes of user’s behaviors on the internal network. Everyone inside network is generally trusted which allows those authenticated users to launch any of the attacks from several different locations and targeted many specific destinations inside the network with devastating results.

Data generated from NIPS tool tends to have very huge volume, and asymmetrical, making the performance of traditional analysis algorithms inefficient and lead to exhaust the processor.

Therefore, using data mining algorithms with improvement k-mean of a cluster algorithm can take advantage of high-performance computers that compute and perform tracing for analysis data network generated from Snort, which is considered as one of the NIPS tools. Improving the intrusion detection within the internal network and generating rules to prevent this intrusion in the future is a main component of this study.

To detect known attacks, our approach will use the Snort which is an NIPS based signature technique. While unknown attacks will be detected using two sub-phases of the data mining approach integrated with the Snort, this is also anomaly inspection technique but it the probability of identify new attacks is low in anomaly detection. Moreover, the Snort is unable to detect internal network scans because it does not examine like most security

software and intrusion detection. While the proposed approach of data mining helps to overcome these challenges and significantly improve in the ability detection of Snort.

5.2. Conclusion

Improve the work of the Snort integrated with data mining methods in implementation mechanism of an anomaly- based and signature-based prevention system effectively, and the installation of this system as an internal gateway for the network, this system will contribute to improvement significantly in the following:

1. The strength of any network security system lies on its weakest point. Although it can be impossible to protect against all types of attacks, such an individual can implement, the improved NIPS base on both misuse and anomaly detection approach can expose such an individual and lead to prevent before more damage is done.

2. Cooperation between the data mining and Snort helps to reduce the analysis effort of the NIPS when high- network records availability. This will lead to achieve minimized latency.

3. The data mining approach is complementary to the Snort that will help to increase overall attack coverage and specially insider attack. The existence of the proposed centralized system reduces the load of the intrusion prevention tasks on every individual host.
4. Maximizing level of security network for all hosts inside the network from internal threats as well as external threats.
5. The proposed system will have a visualization interface to help the analyst in better understanding suspicious behavior traffic detected by the data mining sub-phases then take the appropriate decision will reduce human mistakes.
6. Using centralized database to store all network records and analyze it by the data mining sub-phases. Detecting a new attack or any internal threat, will update attack file in a centralized proposed system. This helps to solve difficulties in deploying the mobile agents on every host and also solve a problem update in each host.
7. The anomaly detection approach represented by two data mining sub-phases is effective for detecting many insider attacks, where authorized user attempts access to source with the attack aim. The malicious behavior shown by such a user is often at a difference with normal behavior, and can be selected as anomalous behavior. Since no security mechanism is fully guaranteed and undetected successful insider attack creates equivalent to outsider attack dangerous. Using our approach to analysis and to detect insider attack will increase a power of security from inside network. This makes your network as a secure fortress from the inside and outside.

8. Using two sequential analysis stages that represented by two data mining sub-phases will achieve increases the accuracy of the analysis and reduces the effort on human analyst and on NIPS. In addition, it effectively helps in detection hidden malicious paths between source and destination. Therefore, it will achieve to reduce a rate of false positives, and false negatives alarms.

9. Because our primary goal is to discover the process of suspicious reconnaissance within the network before launch any network attacks with devastating results. Therefore, the focus was on the use of IP source, port source, IP destination, and port destination as main attributes of network records that will be the analysis in our approach is more suitable and efficient in the analysis. In addition using these specific attributes leads to accuracy in the results, minimizes consumption time and process time of analysis, reduces overload even when huge network records availability for analysis.

5.3. Future work

In the future, we would like to expand and develop this work in the following directions:

First, it will improve the execution of new rules generated by the analysis of data mining sub-phases. By attempting implementation of these new rules through the Snort files rather than executing it through Netsh command- line.

Second, Increase the accuracy of the cluster's classifications by adding other attributes for analysis network records through the data mining sub-phases, such as the connection time it takes between the source and destination with taking into account non increase the overload process and time.

Third, Increase improvement of the whole system in terms of accuracy and efficiency by using Genetic Algorithm (GA) is known one of Artificial Intelligence algorithms to generate effective solutions for optimization and search problems. It will use the Genetic Algorithm as the independent sub-phase working in parallel with the data mining sub-phases and as this system will be composed of a batch of sub-phases. The attack is expected to have the alarm generated from both Genetic Algorithm (GA) based sub-phase, and data mining based sub-phases. Therefore, the false alarm may be eliminated due to the unified decision on the attack must be from both these sub-phases.

Finally, it is possible to upgrade the proposed system to be an automatic one with minimum human intervention. A smart system will be able to read sniffer data, distributing these data on appropriate sub-phases depending on their work, analysis of the data and make appropriate decisions without reference to the analyst, and generation of smart rules to prevent any type of threats. All these actions will be automatically without human control.

References

[1] Kizza J.M, “A Guide to Computer Network Security”, Computer Communications and Networks, chapter 13 System Intrusion Detection and Prevention, Springer-Verlag London Limited, 2009.

[2] Lucas Tamagna-Darr, “Evaluating the Effectiveness of an Intrusion Prevention / Honeypot Hybrid”, Master thesis, Rochester Institute of Technology , B. Thomas Golisano College of Computing and Information Sciences Department of Network Security and Systems Administration ,August, 2009.

[3] Alaa H. Al-Hamami, “Data mining: concepts, techniques and application”, Ithraa publishing and distribution, Amman, Jordan, 2008.

[4] TERRY BRUGGER S, “Data Mining Methods for Network Intrusion Detection”, PhD thesis, University of California Davis ,2004.

[5] Charalampos Zois,Herbert Bos, ”Intrusion Prevention System” ,Vrije universities,2006.

[6] Dinesh Sequeira , “Intrusion Prevention Systems- Security-Silver Bullet?” , GSEC Version 1.4B,OPTION 1,SANS Institute Reading Room site 2002.

[7] V´aclav Sn´aˇsel, Jan Platoˇs, Pavel Kr¨omer and Ajith Abraham, “Matrix Factorization Approach for Feature Deduction and Design of Intrusion Detection Systems”, The Fourth International Conference on Information Assurance and Security, IEEE, 2008.

[8] Andrés, Steven; Kenyon, Brian, Security Sage’s Guide to Hardening the Network Infrastructure, Understanding Intrusion Detection and Prevention Basics. Rockland, MA, USA: Syngress Publishing, 2004.

[9] “How Data Mining is Used for Intrusion Detection”, spam laws, http://www.spamlaws.com/how-data-mining-helps-intrusion-detection.html,time access 1/4/2010.

[10] Cheung-Leung Lui , Tak-Chung Fu , Ting-Yee Cheung , “Agent-based Network Intrusion Detection System Using Data Mining Approaches”, the Third International Conference on Information Technology and Applications,IEEE,2005.

[11] Theodoros Lappas and Konstantinos Pelechrinis ,Data Mining Techiques for (Network) Intrusion Detection systems , Department of Computer Science and Engineering UC Riverside, Riverside CA 92521, 2007.

[12] Chang-Tein Lu, Arnold P.Boedihardjo, Prajwal Manalwar, Exploiting Efficient Data Mining Techniques to Enhance Intrusion Detection Systems, IEEE ,2005.

[13] Association Rules - Introduction, http://www.resample.com/xlminer/help/Assocrules/associationrules_intro.htm, access time is: 9-10-2010.

[14] Desheng Fu, Shu Zhou and Ping “The Design and Implementation of a Distributed Network Intrusion Detection System Based on Data Mining”, World Congress on Software Engineering, IEEE, 2009.

[15] Evgeniya Nikolova, Veselina Jechev, “Anomaly Based Intrusion Detection Using Data Mining and String Metrics”, International Conference on Communications and Mobile Computing, IEEE,2009.

[16] Pablo Garc´ıa Bringas and Yoseba K. Penya, “Next-Generation Misuse and Anomaly Prevention System”, LNBIP 19, pp. 117–129, 2009.

[17] Jiong Zhang, Mohammad Zulkernine," Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection", International Conference on Intelligent Computation Technology and Automation, IEEE, 2006.

[18] Wang Xuren and He Famei, “Improving Intrusion Detection Performance Using Rough Set Theory and Association Rule Mining”, International Conference on Hybrid Information Technology, 2006.

[19] Saadat Malik, Network Security Principles and Practices, Chapter 14. What Is Intrusion Detection, Cisco Press, 2002.

[20] Kaled Hussain Azrane, “Analyze the Delay Time by Data Mining for Network Intrusion Prevention System using Bro”,Master thesis,Universiti Utara Malaysia ,College of Arts and Sciences(Applied sciences), April, 2009.

[21] puketza,N."Approches to Computer Security: Filtering ,Testing and Detection.University of california Davis,2000.

[22] Andreas Fuchsberger, ”Intrusion Detection Systems and Intrusion Prevention Systems”, Information Security Technical Report, Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX,United Kingdom, Published by Elsevier Ltd. 2005.

[23] Najib Bin Limun, THESIS “the use of Intrusion Prevention System to increase computer security”, Universiti Teknologi Mara, November 2005.

[24] Kumar A, V, Lazarevic, P. Dokas, L. Ertoz,J. Srivastava ,”Data Mining for Network Intrusion Detection”,Research supported by Army High Performance Computing Research Center Department of Computer ScienceUniversity of Minnesota ,2002.

[25] Miller, M,”Learning cost-sensitive classification rules for network intrusion detection using ripper”,Computer Science Department, Columbia University ,1999.

[26] Daniel Barbard, Julia Couto, Sushil and Jajodia Ningning Wu “ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection”, George Mason University ,Center for Secure Information Systems,2001.

[27] Jianxiong, L, Bridges SM ," Mining Fuzzy Association Rules and Fuzzy Frequency Episodes for Intrusion Detection ,2000.

[28] Chris Sinclair ,Lyn Pierce ,and Sara Matzner,"An Application of Machine Learning to Network Intrusion Detection",Applied Research Laboratories The University of Texas at Austin,1999.

[29] Richard P. Lippmann , Robert K. Cunningham ,"Improving intrusion detection performance using keyword selection and neural networks" ,2000.

[30] Susan M. Bridges , Rayford B. Vaughn ,"Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection ",2000.

[31] WENKE LEE, SALVATORE J. STOLFO, "A Framework for Constructing Features and Models for Intrusion Detection Systems",2000.

[32] Wei Fan , Wenke Lee , Salvatore J. Stolfo , Matthew Miller ,"Multiple Model Cost-Sensitive Approach for Intrusion Detection",2000.

[33] Ramesh Agarwal , Mahesh V. Joshi ,"PNrule: A New Framework for Learning Classifier Models in Data Mining (A Case-Study in Network Intrusion Detection) ",2000.

[34] Vipin Kumar,Jaideep Srivastava "Data Mining for Rare Class Analysis",Project for network intrusions and security breaches, Computer Science Department ,University of Minnesota ,2003.

[35] Srinivasulu, P, D Nagaraju, P Ramesh Kumar, and K Nageswara Rao,"Classifying the Network Intrusion Attacks using Data Mining Classification Methods and their Performance Comparison ",IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.6, June 2009.

[36] LEE, W. AND STOLFO, S. J. "Data mining approaches for intrusion detection". Computer Science Department, Columbia University, 1998.

[37] Salvatore J. Stolfo , Wei Fan , Wenke Lee , Andreas Prodromidis , Philip K. Chan "Cost-basedModeling for Fraud and Intrusion Detection: Results from the JAM Project",2000.

[38] Guy Helmer , Johnny S. K. Wong , Vasant Honavar , Les Miller "Automated discovery of concise predictive rules for intrusion detection", Department of Computer Science, Iowa State University, Ames, IA.,2001.

[39] Kenji Yamanishi , Jun-ichi Takeuchi , Graham Williams ,"On-line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms ",2000.

[40] Terran Lane , Carla E. Brodley ,"Temporal Sequence Learning and Data Reduction for Anomaly Detection",Purdue University,ACM Transactions on Information and System Security,1998.

[41] Jake Ryan , Meng-jang Lin , Risto Miikkulainen ,"Intrusion Detection with Neural Networks",1998.

[42] Kokkinaki,A. I, ”On Atypical Database Transactions: Identification of Probable Frauds using Machine Learning for User Profiling”, IEEE Knowledge and Data Engineering Exchange Workshop,1997.

[43] Wei Fan , Matthew Miller , Salvatore J. Stolfo , Wenke Lee,"Using Artificial Anomalies to Detect Unknown and Known Network Intrusions",IEEE International conference on Data Mining,2001.

[44] Kingsly Leung,Christopher Leckie,"Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters",2005.

[45] S. Ramaswamy, R. Rastogi, and K. Shim,” Effcient algorithms for mining outliers from large data sets”, 2000.

[46] Markus Breunig , Hans-Peter Kriegel , Raymond T. Ng , Jörg Sander ,"LOF: Identifying Density-Based Local Outliers ",2000.

[47] ShengYi Jiang,Xiaoyu Song ,Hui Wang,"A clustering-based method for unsupervised intrusion detections",2006.

[48] Tamas Abraham,"IDDM: Intrusion Detection using Data Mining Techniques",Information Technology Division Electronics and Surveillance Research Laboratory,2001.

[49] Daniel Barbará , Julia Couto , Sushil Jajodia , Leonard Popyack , Ningning Wu,"ADAM: Detecting Intrusions by Data Mining",IEEE ,2001.

[50] Barbara,D, S Jajodia, N Wu,"Mining Unexpected Rules in Network Audit Trails",1999.

[51] Levent Ertoz . et Al, "MINDS - Minnesota Intrusion Detection System,", Next Generation Data Mining Chapter 3, 2004.

[52] Ambareen Siraj, Rayford B. Vaughn, and S. M. Bridges, "Intrusion Sensor Data Fusion in an Intelligent Intrusion Detection System Architecture”, 2004.

[53] http://www.rising-global.com/, Accessed on 15/2/2011.

[54] http://www.snort.org/, Accessed on 20/2/2011.

[55] BuKisa share you knowlage ," How an Intrusion Prevention System Works", Mar 22nd, 2011 by Lexus, http://www.bukisa.com/articles/472297_how-an-intrusion-prevention-system-40ips41-works,Accessed on 20/2/2011.

[56] Kris Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems", Chapter 9 Probes, Master's Thesis, Massachusetts Institute of Technology, 1999.

[57] WinPcap http://www.winpcap.org/ last access in May 3, 2011.

[58] Nalneesh Gaur, Snort: Planning IDS for your enterprise, published by linux journal in Jul 11, 2001 http://www.linuxjournal.com/article/4668?page=0,0 last access in May 3, 2011.

[59] Rafeeq Ur Rehman, "Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, MySQL, PHP, and ACID", ISBN-10: 0131407333, Publisher Prentice Hall, 2003.

[60] Li Tian, Wang Jianwen," Research on Network Intrusion Detection System Based on Improved K-means clustering algorithm", IEEE, 2009.

[61] M. Zaki and W. Meira Jr. Fundamentals of Data Mining Algorithms, Cambridge, 2010.