ISSN 2348-375X

Unique Journal of Engineering and Advanced Sciences

Available online: www.ujconline.net

Research Article

A SCALABLE AND HYBRID SECURITY SYSTEM FOR DAMAGE CONTAINMENT

Kaviya S[1*], Uma S[2], Krishnapriya KV[3], Nandhini N[4]

[1]Computer Science and Engineering, Hindusthan Institute of Technology, Coimbatore, Tamilnadu, India
[2]Professor and Head of the PG Department of Computer Science and Engineering, Hindusthan Institute of Technology, Coimbatore, Tamilnadu, India
[3]Computer Science and Engineering, Hindusthan Institute of Technology, Coimbatore, Tamilnadu, India
[4]Computer Science and Engineering, PM college of engineering, Krishnagiri, Tamilnadu, India

Received: 24-01-2015; Revised: 22-02-2015; Accepted: 20-03-2015

*Corresponding Author: **Kaviya S**
Computer Science and Engineering, Hindusthan Institute of Technology, Coimbatore, Tamilnadu, India

ABSTRACT

Intrusion Prevention is the ability to provide adequate facility to detect security threats and protect the intended System. The Security System is to be designed such that the damage caused by the intruder is contained and, perhaps its progress stopped. The Premise of intrusion prevention is that to secure the system operation before an intrusion actually occurs proactively, rather than detecting attacks only in progression. But certain attacks (namely unknown attacks) can be handled only at the time of action reactively. This type of monitoring ensures stopping of attacks before completion automatically.

A Distributed Scheme, based on sharing information between trusted peers in a network to guard the network, as a whole against intrusion attempts can be employed. A **Peer-to-Peer** infrastructure is used to distribute up-to-date rumors, facts, and trust information in a scalable manner. The Security System should be capable of defeating any type of thwart. This necessitates both **Anomaly based approach** and **Protocol misuse detection**. This in turn requires or involves generation of audit data in response to access requests, which records sufficient information to establish what event occurred and what caused the event. This type of formulation makes the **Security System both Scalable and Hybrid.**

The Principle behind anomaly detection is to employ **Distance-based metric algorithm** on clustered dataset. This Clustering is done on the unlabeled data on which the unsupervised anomaly detection is performed. The basis of misuse detection (data mining based) lies in the use of **Learning algorithm** applied on the labeled dataset. We also propose an adaptive threshold to change its threshold level of identification of abnormal activity based on the number of false negatives and false positives.

The Security System is needed especially when it is identified what the attacker is looking for. Then, after each attempt by the attacker, the Security System should go and see if the System in question is vulnerable or not.

General Terms – Security, Protocol, Intrusion, Legitimate, Misuse, Meta Data, Clustering

Keywords: Peer-to-Peer, Anomaly and Misuse detection, False Negatives and False Positives.

INTRODUCTION

We are interested in automated capabilities that can detect or find anomalies in computer systems that report them in useful ways and remove discovered anomalies. One uses intrusion detection tools to watch what is going on in the network to discover suspicious events. If perfect intrusion detection and reaction systems were available, there might be no need for any other measures to protect against **Cyber attack.**

Prudent, affordable, continuous protection of one's network involves monitoring the network for anomalies of various kinds, whether they are suspicious textual strings in a network packet or undesirable values. Moreover, it involves correcting detected anomalies, whether that means terminating a connection or reconfiguring a server.

We call an automated system that performs or assists in such tasks a Security system. Besides checking network packets for suspicious strings, or monitoring a user's behavior looking for deviations from an established pattern, the system checks components of the network for errors of omission, misconfigured applications, and errors in system parameters. When the system finds an anomaly, it reacts, generally by trying to fix the anomaly. Its response may be restricted to issuing an alert for certain anomalies. For others, it may be able to fully correct the problem. In some cases, it may be able to provide ancillary information that will assist an administrator in correcting the anomaly. The Premise of intrusion prevention is that to secure the system operation before an intrusion actually occurs proactively, rather than detecting attacks only in progression. But certain attacks (namely unknown attacks) can be handled only at the time of action reactively. What it can do will be determined by the state of the art and the information operation to be protected. The Security System should be capable of defeating any type of thwart. This necessitates both anomaly based approach and protocol misuse detection[1].

The extent of the protection domain determines the needed capacity of the Security system for that domain. Networks tend to grow, thereby extending the scope of interest for a Security system. A Distributed Scheme[2], based on sharing information between trusted peers in a network to guard the network, as a whole against intrusion attempts can be employed. A **Peer-to-Peer**[3] infrastructure is used to distribute up-to-date rumors, facts, and trust information in a scalable manner. Thus, scalable Security systems are needed, not only so that the same basic system can serve domains of different size, but also so that it can accommodate significant growth in the domain it protects. Data-level security is entailed as a blend of policy and encryption. Encrypting data where it resides and as it travels across the network is indispensable because, if all other security measures fail, a strong encryption scheme protects the proprietary data. Thus this type of formulation of the **Security System,** which is **both Scalable and Hybrid**[3]**,** is being proposed.

Organization of the Paper

This paper is organized as follows: First, we describe our model and give a brief discussion of methods of intrusion detection. We follow this by describing the components of our system and their central workings. In Section 4 we describe the Architecture of Security system. In Sections, 5, we prove our concepts and suggest future works.

Overview of the Model

The two flavors of Attack Detection are Anomaly and Misuse Detection.

- **Signature-Based**, designed to match 'signatures' of attack/intrusion from 'pre-installed' signature database. This means that these types of systems can only detect 'known' attacks. A weakness is that attacks cannot be detected until corresponding attack signatures are entered into the knowledge database.
- **Anomaly-Based**, designed to detect novel attacks/intrusions. Such approach typically achieves this with self-learning. Anomaly techniques assume that intrusions display anomalous characteristics that render them detectable. Given a "normal activity profile," intrusions produce behavior that varies from an established historical profile by statistically relevant amounts. To be successful, an anomaly detection method must first be able to identify abnormal behavior.

1. Hybrid and Smart Detection Technique

Our Model employs both the Approaches as in Fig. 1, to be most accurate and reliable against cyber intrusions.

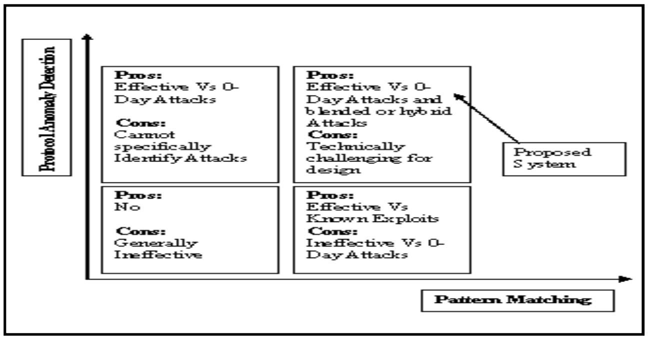

Protocol Anomaly Detection
Pros:
Effective Vs 0-Day Attacks
Cons:
Cannot specifically Identify Attacks
Pros:
Effective Vs 0-Day Attacks and blended or hybrid Attacks
Cons:
Technically challenging for design
Proposed System
Pros:
No
Cons:
Generally Ineffective
Pros:
Effective Vs Known Exploits
Cons:
Ineffective Vs 0-Day Attacks
Pattern Matching

Figure 1: Comparison between Pattern Matching and Protocol Anomaly Detection

Taking into consideration the **Temporal** and **Spatial** aspects to prevent high false positive rates as in Fig 2, behavior-based intrusion detection systems must recognize the time variant characteristics of the environment. Discernible patterns may only appear in analysis using several different metrics, possibly spread over multiple time regions. Our objective is to understand these nuances and to richly describe the notion of behavior across multiple time granularities to reduce false positives. Because attack behavior occurs irregularly over time, it is essential to analyze groupings of sessions for malicious behavior.

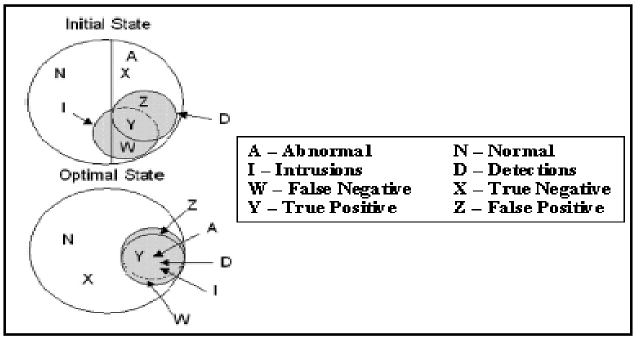


Figure 2: Normal, Abnormal, and Malicious Events

2. Advanced Management Reducing False Positives using

▪ **Fine Grained Filtering**

This is can be done by combining a broad set of filtering parameters such as IP address, port, alert type etc with Boolean Logic. Allowing filtering to occur appropriately at each layer, resulting in more efficient event/alert processing model

▪ **High Level Correlation**

While each discrete event may not qualify as a malicious or even suspicious activity, correlating a combination of events may results in a high fidelity alert

▪ **Unsupervised Self Learning**

Misuse Detection model can be retrained by adding labeled instances of the new attacks into the dataset and the method would readjust its rule set to detect them.

Anomaly Detection model can detect new types of intrusions as they will deviate from the normal network usage. It can clusters unlabeled data instances into clusters based on distanced based metric.

3. Trusted Prevention
Multi Dimension Prevention

- **Packet Scrubbing**
  If parts of the packet are malicious this method has the ability to rewrite the offending part to something non-malicious.
- **TCP Wrapper and Session Snipping**
  This is used to sand box the TCP and UDP services.
- **Host based Firewall**
  Upon the detection of an attack, IDS sends a notification to the firewall to block offending traffic, passing along specific parameter for blocking.
- **Proactive Information Alert**
  Upon the detection of an attack, IDS sends a alert information to all its peer in the Secured Network as in Fig 3., passing secured data with hash functions, digital signatures.

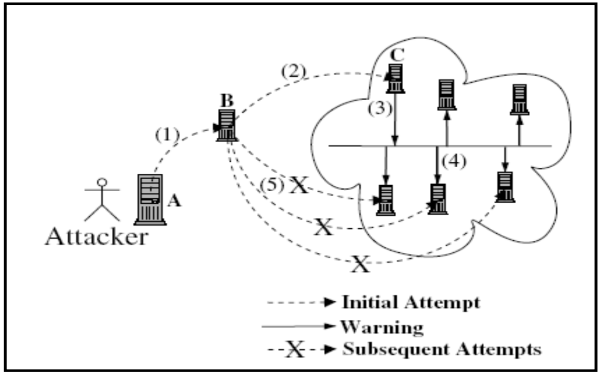


Figure 3: Proactive Alert

4. Architecture
Local Sub-System

- **Prevention Module**
  - Packet Scrubbing
  - TCP Wrapper Programs
  - Host-based Firewall
- **Detection Module**
  - Network Monitor
  - Intrusion Scanner
  - Attack Resistor
- **Statistical Analysis**
  - Measure Classification
  - Algorithm for Comparison
  - Long-Term Profile Training and Update
    - Level I: Training – Expected Behavior
    - Level II: Deviations
    - Level III: Thresholds for Measure
- **Protocol Analysis**
  - Event Abstraction Module
  - Pattern Matching Module
- **Decision Module**
  - ADAM Module
    - Transition based Auditing – Safe State of the System
  - Audit Data Management
  - Reporting Module
- **Intrusion Detection Information Base**
  - Audit Databases
    - OS Audit Records
    - Detection Specific Audit Records
- Subject
- Action
- Object
- Exception-Condition
- Resource Usage
- Time-Stamp
- **Remote Sub-System**

<table>
<tr><th colspan="4">Prevention Module</th><th colspan="6">Detection Module</th></tr>
<tr><td rowspan="3">Packet Scrubbing</td><td colspan="2" rowspan="3">TCP Wrapper Programs</td><td rowspan="3">Host-based Firewall</td><td colspan="2">Network Monitor</td><td colspan="2">Intrusion Scanner</td><td colspan="2">Attack Resistor</td></tr>
<tr><td colspan="3">Protocol Analysis</td><td colspan="3">Statistical Analysis</td></tr>
<tr><td>Pattern Matching</td><td colspan="2">Event Abstraction</td><td>Measure Classification</td><td>Comparison</td><td>Profile Training</td></tr>
<tr><td colspan="10">Decision Module</td></tr>
<tr><td colspan="3">ADAM Module</td><td colspan="3">Audit Database Management</td><td colspan="4">Reporting Module</td></tr>
<tr><td colspan="10">Intrusion Detection Management Information Base</td></tr>
<tr><td colspan="4">OS Audit Records</td><td colspan="6">Detection Specific Records</td></tr>
<tr><td colspan="10">Remote Sub-System Interface</td></tr>
<tr><td colspan="4">Enlargement of MIB</td><td colspan="6">Propagation from MIB</td></tr>
<tr><td colspan="2">Decryption</td><td colspan="2">Data Update</td><td colspan="3">Encryption</td><td colspan="3">Data Send</td></tr>
</table>

Figure 4: System Architecture

Detection Module

Network Monitor

To capture the packets that are destined to that host & also the broadcasts and to prevent the networks attacks such as

1. ACK Storm
2. Denial of Service
3. Distributed Denial of Service.
4. Data Diddling
5. Repudiation
6. Masquerading or Spoofing

IPSec (IP Network Security) provides the capability to secure communication across a LAN. It provides security services at the IP Layer by enabling a system to select security protocols. The advantage is that it is transparent to the end users and application and provides a general purpose solution. It includes a filtering capacity so that only selected traffic need incur the overhead of IPSec processing.

Intrusion Scanner

To identify the type of event that is captured through the Network Monitor. It has to categorize the event using Anomaly and Misuse Detection techniques. It looks for evidence of infraction, including intrusions by outsiders and violations of policy by insiders. It maintains the session in a detection window by monitoring the events until it decides the event to be normal or malicious.

It maintains logs for all the successfully completed events that are normal but not classified by the Security System throughout its execution time. Later (when the occurrence of a set of events exceeds the threshold) it updates the signature Information base to record the normality of the captured events. Attack events are directly passed on to the decision module[4].

Protocol Analysis

Protocol flow analysis gives the rudimentary knowledge of a particular application protocol. With this knowledge it decides what part of a protocol to inspect, if any part at all. This significantly reduces processing time, since it would usually inspect the ignored protocol data with all the applicable rules. Flow analysis also reduces false positives by limiting the amount of inspection that it does, so it is less likely to alert on unrelated rule content matches.

Pattern Matching

Wu-Manber Algorithm[5]

The algorithm starts by pre-computing two tables, a bad character shift table, and a hash table. When the bad character shift fails, the first two characters of the string are indexed into a hash table to find a list of pointers to possible matching patterns. These patterns are compared in order to find any matches and then the input is shifted ahead by one character and the process repeats.

Overall the algorithm achieves a worst-case performance that is no better than naive string matching, but the average case performance is among the best of all multi-pattern string matching algorithms. In the worst case the algorithm requires for every character of input a memory access to the shift and hash table, followed by as many string compares as there are patterns to be matched (this can only happen if the hash fails). The algorithm is widely used and achieves very good average-case performance.

For Pattern “this” and Hypothetical_K=3,

- xthxxis
- txhxixs
- thxxixs
- thixxxs
- txhxxis
- xxthxis

All the above input-sequences can be matched by the Wu-Manber Algorithm. The Protocol Rules can be symbolized using alphabets and stored. This is a sort of preprocessing. In the above example “this” is a symbolized protocol rule.

The Steps involved in the Protocol Analysis are given below:

```
Procedure {Protocol-Analysis}
Input-Sequence ← Load from Detection Window
Hypothetical_K ← User Specified Threshold
Identify the Protocol
Rule-Set ← Identified Protocol’s Rule Cluster
While {Rules in Rule-Set un-flagged}
Construct Character Shift Table
Entry in Hash-Table
Flag Rule in Rule-Set
End While
For each Packet in the Input-Sequence
// Wu-Manber Pattern Matching
Input-Pattern ← Protocol-part of the Packet
Initialize Actual_K ← 0
While {Input-Pattern}
// Match Input-Pattern against Hash and Character Tables
If Match then
Shift One-character in Character Table
Else
Increment Actual_K
End {If Match}
// Compare Actual_K and Hypothetical_K
If (Actual_K > Hypothetical_K)
Value (Input-Pattern) ← Normal
Actual_K (Input-Pattern) ← 0
Else
Actual_K (Input-Pattern) ← Actual_K
End If
End While {Input-Pattern}
End {For each Packet}
Sum up all non-zero values of Actual_K for Warning-Level
Output Warning-Level
End Procedure {Protocol-Analysis}
```

Statistical Analysis

For Statistical Analysis we make use of **Test of Goodness** of Fit using **Chi-Square Distribution** is done by the following formulation,

$$\chi^2 = \Sigma_{i=1 \text{ to } n} (O_i - E_i)/ E_i$$

By this test, we test whether differences between the observed and expected frequencies are significant are not. In the above formula,

Oi – set of observed or experimental values or frequencies

Ei – set of expected or theoretical values or frequencies and

The above Chi-Square Distribution follows with n-1 degrees of freedom and with the condition,

$\Sigma_{i=1 \text{ to } n} O_i = \Sigma_{i=1 \text{ to } n} E_i$

```
Procedure {Statistical-Test}
// Initialization
Look-Up ← Load $\chi^2$ Table Values
Test-Data ←Event-Log // after Preprocessing
Hypothetical-Data ← Load from Data-Store // after Preprocessing
Compare Test and Hypothetical Data
// The application of the Chi-Square Distribution basis formula
Actual_Value ← result of Formula application
If {Actual_Value < Look_Up} then
Accept $H_0$ and conclude the difference is not significant
Else
Reject $H_0$ and conclude the difference is significant
End if
Return {Accept or Reject based on H0}
End Procedure {Statistical-Test}
```

Conditions:

- Frequency N should be greater than 50, reasonably large
- Both Normal and Attack data are used for conclusion

- One calculation uses the mean of the χ^2 from the normal data and the other uses the mean of the χ^2 from the attack data
- The appropriate calculation is compared against the Upper Control Limit.

The Upper Control Limit is calculated using the accumulated normal profile statistics sample mean and adding twice the normal profile statistics standard deviation,

Upper Control Limit =X + 2S.

Adaptive Threshold is set using the Standard deviation using the formula,

$\boldsymbol{\sigma = \sqrt{} \Sigma\ (x-m)^2 / (n-1)}$

σ – Standard Deviation

n – Number of samples

(x-m) – Value of current sample minus mean, Threshold is simply the mean plus the standard deviation.

The **Locality Frame**[6] is a value determining the length of a temporally local region over which the number of mismatches is summed up. For example, if the locality frame is set to 20, then at each point of the test data the number of mismatches in the last 20 (overlapping) sequences, including the current sequence, is determined. The number of mismatches that occur within a locality frame is referred to as the **locality frame count** (LFC). The locality frame count is the final value that is used to determine how anomalous the test data is. The length of the locality frame should be a user-defined parameter that is independent of the length of the detector-window used to segment both test data and hypothetical data against which the comparison has to be done.

LFC is bound with both Pattern Matching and Chi-Square Variate. The Value of k in Pattern Matching can be dependent on LFC Parameter. LFC value may also be adaptively changed using Chi-Square Distribution for efficient performance rather than a fixed value. Hence LFC is also an important and critical factor in Pattern Matching.

Attack Resistor

This resets the connection to terminate an attack in progress.

Passive Fingerprinting is a method to learn more about the enemy, without them knowing it. Passive fingerprinting is based on sniffer traces from the remote system. Instead of actively querying the remote system, all it does is capture packets sent from the remote system. It also alert the Security System at the remote host to shut down the malicious process.

Decision Module

ADAM (Audit Data Analysis and Mining)[7] uses data mining to build a customizable profile of rules of normal behavior, and a classifier that sifts the suspicious activities, classifying them into real attacks and false alarms. ADAM is designed to use in real time, a characteristic achieved by using incremental mining algorithms.

ADAM is essentially a test-bed for using data mining techniques to detect intrusions. ADAM uses a combination of association rules mining and classification to discover attacks in a **TCP dump** audit trail. ADAM builds a repository of "normal" frequent item-sets that hold during attack-free periods. It does so by mining data that is known to be free of attacks.

For Data Mining Java-based tool **Weka**[8] **(Waikato Environment for Knowledge Analysis) Version 3-4-3** which provides implementations of state-of-the-art learning algorithms that can be applied on the dataset. It has algorithms for discretization. The learning methods of the tool are called classifiers. The tools for preprocessing the data are called filters. The main focus of Weka is on classifier and filter algorithms.

Prevention Module

This is to prevent the known attacks. The following tools can be used as,

Use of TCP Wrappers: This can be done based on the rules located in the files **/etc/hosts.allow** and **/etc/hosts.deny**. The TCP Wrappers program can log incoming connections via **syslog.** TCP Wrappers can be run as a stand-alone program but nowadays it is most commonly used as a library (libwrap) that is linked to the **inetd** program.

Use of Host-based Firewall Program: This can be done by firewall program such as ipfw, to block access to servers from specific network. Rules for the host-based firewalls are loaded into the Unix kernel when the system boots although it can be fine-tuned during the system operation.

Packet Filtering

- Source Address
- Destination Address
- Specific Service Ports

Port Filtering: Packet Filtering is done based on information available at the TCP Layer.

Access Control Lists

- Source and Destination Address IP
- Type of Connection Agent
- Type of requested Network Service
- User requesting the Connection
- Time and Day
- Encryption

Audit System

Errors Access Permissions – To prevent Masquerader: To detect when someone is randomly attempting to access files in the hope of getting at something that was not properly protected. It reports when a file opened was denied access.

Errors Granted Access – to prevent Misfeasor: To indicate attempts to access files without proper access authorization. It reports the number of times access to files opened successfully was denied.

Errors Supervisory Access – to prevent Clandestine User: To indicate attempts to access audit files without permission, prevent or suppress audit collection, seizure of supervisory control. It reports the number of attempts from Non-supervisory Users to evade the Security System that were successfully denied.

Errors Logon: To indicate the number of failed logon attempts to the server.

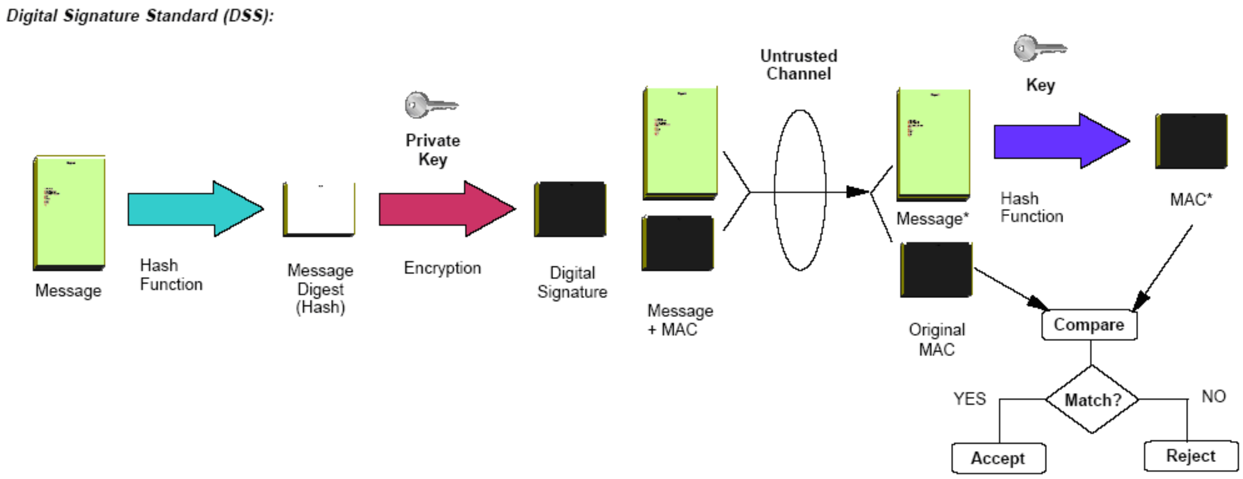
Call Back: Call back is to provide an important service in secure environment. It calls a user back at a predefined address and prevents intruders who have obtained valid logon info from dialing at an unauthorized location.

Remote Sub-System

Issues that need to be addressed if cooperative intrusion detection using data sharing between distinct systems becomes a viable option, and provide a set of requirements for designing such a system. A formal model meeting these requirements is to be implemented as a functional cooperative data-sharing system[9].

- Share encrypted security information with any host in the LAN
- Powerful searchable database of relevant security data
- Detects light but network-wide attacks
- Keeps historical data of system status

A Peer-to-Peer Infrastructure is used to distribute up-to-date rumors, facts, and trust information in a scalable manner. The broad goal is to distribute attack information (gathered by the intended victim) among all peers in a P2P network. . The chance that at least one of the machines does notice an attack before the actual attack occurs on that particular machine. This increases level of security of the entire P2P network. This makes it very attractive to have a system spreading such information quickly and widely.



SUMMARY AND FUTURE WORKS

While the types of attacks are changing and increasing, this approach is capable of constantly ensuring the Network protection, by providing up to date signatures. The customizable engines provide additional strength and flexibility to alert and defend against the mounting number of threats. Armed with a thorough understanding of the capabilities of the Security System, an administrator has a powerful tool with which the malicious attackers can be kept at bay. The Security system is needed especially when it is identified what the attacker is looking for.

The measure for the actual performance of the Security System has to consider the False Positives, False Negatives, Overhead due to IP Layer Security and Efficiency of the Learning Algorithms. To improve system performance and extensibility, each producer and consumer runs as an individual Java Threads (light weight process) in the same JVM. To achieve maximum efficiency of the inter thread communications, the event buffer is implemented by shared memory. The Threads are synchronized using semaphores.

The overall objective of the Security System is as follows:

- Broad Detection Range – the ability to distinguish Intrusions from Normal Activities
- Economy in Resource Usage – the efficiency of using System resources
- Resilience to Stress – the ability to function correctly under high load

REFERENCES

1. Tysen Leckie, Alec Yasinsac, Metadata for Anomaly-Based Security Protocol Attack Deduction, IEEE Transactions On Knowledge And Data Engineering, 2004; 16: 9.
2. Joglekar Sachin P, Tate Stephen R, ProtoMon: Embedded Monitors for Cryptographic Protocol Intrusion Detection and Prevention, Department of Computer Science and Engineering, University of North Texas, Denton, TX 76203.
3. Kymie MC. Tan, Kevin S. Killourhy, and Roy A. Maxion , Undermining an Anomaly-Based Intrusion Detection System Using Common Exploits", Dependable Systems Laboratory, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213, USA.
4. Nathan Tuck, Timothy Sherwood, Brad Calder, George Varghese, Deterministic Memory-Efficient String Matching Algorithms for Intrusion Detection", Department of Computer Science and Engineering, University of California, San Diego, Department of Computer Science, University of California, Santa Barbara.
5. Alden H. Wright, Approximate String Matching Using Within-Word Parallelism" Department of Computer Science, University Of Montana, Missoula, MT 59812, U.S.A.

6. Josue Kuri, Pattern Matching Techniques in Intrusion Detection, Mastère RIT 98 – 99, SSI - Supélec Rennes.
7. Daniel Barbara, Julia Couto, Sushil Jajodia, Leonard Popyack, Ningning Wu George, ADAM: Detecting Intrusions by Data Mining, Mason University Center for Secure Information Systems and ISE Department Fairfax, Proceedings of the 2001 IEEE Workshop on Information Assurance and Security.
8. Graham Williams, Markus Hegland and Stephen Roberts, A Data Mining Tutorial, Presented at the Second IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN'98), 14 December 1998.
9. Rebecca Cathey, Ling Ma, Nazli Goharian, and David Grossman, Misuse Detection for Information Retrieval Systems, Information Retrieval Laboratory, Department of Computer Science Illinois Institute of Technology, Chicago, IL 60616.

Source of support: Nil, Conflict of interest: None Declared