Speedy Signature Based Intrusion Detection System Using Finite State Machine and Hashing Techniques

Utkarsh Dixit[1], Shivali Gupta[2] and Om Pal[3]

**[1] School of Computer Science, Centre for Development of Advanced Computing,
Noida-201307, India**

**[2] School of Computer Science, Centre for Development of Advanced Computing,
Noida-201307, India**

**[3] Senior Technical Officer, Centre for Development of Advanced Computing,
Noida-201307, India**

Abstract

This paper proposes a secure system designs for client-server based communication systems. In this system, security services are implemented on server, as generally data received on the servers contains malicious contents. The technique that we used is to perform speedy intrusive signature matching received inside a network with the known signatures from the training database. Probable intrusive signatures, which get filtered from hash value matching, are exposed to a finite state model that inspects those signatures against a finite automaton. Other systems like anomaly based detection may not detect all malicious activity signatures. Also, we have taken a note to reduce the false positive rate to nil while implementing the system which gets generated in other detection systems during the communication process. The proposed system works on a client-server based model.

Keywords-*Signature matching, Finite State Machines, Hashing, Host based IDS, Mid-Square Method*

1. Introduction

With the increase in the number of users and the type of facilities they require, the number of computers and in turn the number of networks has also increased drastically. And as there always exists some shortcomings with every technology, network security is also not untouched with it. Thus, network intrusion presents a serious problem for network security as malicious users always look forward to disrupt the services and cripple the capability of the network. Though other methods like Windows firewall, Virtual Private Network or various encryption techniques exists which offer network security, but they have limited capability as they are somewhat static in nature. Intrusion detection is the process of monitoring the events occurring in a computer system or network and analysing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. Intrusion detecting system, for the purpose of dealing with IT, can be categorized broadly into two categories:

Network based IDS: is an independent platform that identifies intrusions by examining network traffic and monitors multiple hosts. Network intrusion detection systems gain access to network traffic by connecting to a network hub, network switch configured for port mirroring, or network tap. In a NIDS, sensors are located at choke points in the network to be monitored, often in the demilitarized zone (DMZ) or at network borders.

Host based IDS (HIDS): It consists of an agent on a host that identifies intrusions by analysing system calls, application logs, file-system modifications (binaries, password files, capability databases, Access control list, etc.) and other host activities and state. In a HIDS, sensors usually consist of a Software agent. Some application-based IDS are also part of this category.

Intrusion detection activity is classified into two categories:

Anomaly Detection: is the attempt to identify malicious traffic based on deviations from established normal network traffic patterns.

Misuse Detection: is the ability to identify intrusions based on a known pattern for the malicious activity.

In this work, we have used host based signature detection system which uses the received packets to decide the intrusive activity along with exposing the known signatures to Finite State Machines so as to clearly decide the existence of the malicious activity into the system. The goal of the research is to analyse various signature based techniques of Intrusion Detection so that a robust system can be developed.

Rest of paper is organized as follows: in next section related work is presented. In Section 3, proposed scheme is given. Section 4 concludes the paper and in last reference section is given.

2. Related Work

James P. Anderson [1] studied the purpose of improving the computer security auditing and surveillance capability of the customer system. Mohammadreza Ektefa [2] proposed to use data mining techniques including with classification tree and support vector machines for intrusion detection. Kingsly Leung [3] introduced grid based and density based cluster approach for separating frequent item sets from non frequent item sets and they used their density based support mechanism for identifying unseen attack in anomaly detection. [4] R. Li, and W. M. Pan, introduced sequence based anomaly detection approach. [5] Karen Scarfone & Peter Mell discuss various issues regarding Intrusion detection and prevention systems. In this paper, proposed scheme uses combination of hashing technology for faster detection of intrusion with finite state machine for reduction of false alarm. It uses the Mid Square Method for implementing hashing. The Mid Square method of hashing can be understood as follows:

Mid Square Method

The middle-square method [6] is a method of generating pseudorandom numbers. The method originated with John von Neumann, and was notably described at a conference in 1949. It is a hash function which uses integer keys values. The mid-square method squares the key value, and then takes out the middle r bits of the result, giving a value in the range 0 to 2^r-1. This works well because most or all bits of the key value contribute to the result. For example, consider records whose keys are 4-digit numbers in base 10. The goal is to hash these key values to a table of size 100 (i.e., a range of 0 to 99). This range is equivalent to two digits in base 10. That is, r = 2. If the input is the number 4567, squaring yields an 8-digit number, 20857489. The middle two digits of this result are 57. All digits of the original key value (equivalently, all bits when the number is viewed in binary) contribute to the middle two digits of the squared value.

Other Hashing Methods:

Modulus Method:

Also known as modulo arithmetic method, it uses

H: Key ----> Integer Index scheme for calculating hash values. E.g. if table size is 100

3 Digit numbers are the keys, 999 possible items, Indices 0..99 on the table

999 % 100 = 99 (100 is Table size), 524 % 100 = 24 etc.

Folding method:

In the folding method, the key is divided into two parts that are then combined or folded together to create an index into the table. This is done by dividing the key into parts where each of the parts of the key will be of the same length as of the desired key. E.g. using the SSN 987, 654, 321, and then add these together to get 1962. We then use either division or extraction to get a three digit index.

Finite State Model:

A Deterministic finite automata is defined by quintuple

D=<Q, ∑, δ, $q_{0,}$ F> where

Q= Finite set of non-empty states

∑= Finite set of input alphabets

The transition function, $\delta : S \times \Sigma \rightarrow S$, where S= Q

Start State, $q_0 \in Q$

A set of accept states, $F \subseteq Q$

3. Proposed Scheme

The overall working strategy is broadly divided into three phases. They are:

(i) Formation of a training database of virus signatures and calculation of their hash values.

(ii) Calculation of hash values of the received data from the client. Thereafter performing speedy signature matching using Mid-Square Method of Hashing

(iii) Implementing a deterministic finite automaton for the filtered signatures for reduction of false positive and false negative alarm.

Phase I:

During the development of the presented system, we analysed the techniques of host based Signature detection system. The developed system is implemented on a server in a client-server environment. The server module maintains a virus signature database file. It also keeps a log of data packets sent by the client. The data received is initially stored in a file in the 15-bit binary format. This format will help resolving collisions that will occur while storing same-length but different virus signatures.

Phase II:

The processing of the received data is commenced by extracting the virus database signatures which are stored in some other file. Signature hash value calculation is done using the Mid-Square technique and stored in a file. This phase takes into account the length of the virus signatures for matching. Comparison of the hash values from the training database was performed easily. The received data, which did not match this first criteria was considered clean and free from intrusion.

Phase III:

The filtered data which passed second phase check was further investigated by exposing the signature strings to the finite state models and depending on the outcomes of the machine the alarms are generated.

Analysis:

The steps we followed in detecting intrusion are as follows:

1. First of all, as a prerequisite a client-server model is established via socket programming, having a client module and a server module.

2. We took into account only that communication which is coming from the client side, i.e. data received from client is stored into a separate file in a 15-bit binary notation.

3. Hashing is applied on both the files, i.e. intrusive signature file and received data file.

4. The initial matching is done against the hash values of the intrusive signature with that of data hash values.

5. Based on the result of the matching, the packets are further investigated or assumed clean in the following manner:

   i) If the hash values are different then we may assume that the data received is free from intrusion and does not require further investigation. This is because our primary criterion is the length of the malicious signature which is stored in our database.

   ii) If the hash values are same then the data needs further investigation.

Thereafter, the data is exposed to a Finite State Module which inspects it for intrusion. We designed finite automata for each defined signature. It reaches to a final state on input of signature strings of known virus only.

To check the working of the system, we sent a known signature named “Ah” from the client to the server.

Taken the signature “Ah”,

Binary equivalent= 100000101101000

Calculated Hash value =7

Received data hash value=7 (we intentionally sent same signature)

Signature matched successfully in second phase. After that data is then put on the automata as follows:

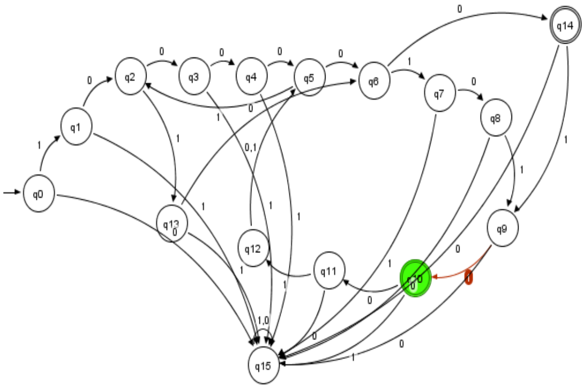

Fig1.Single Finite state model of the several intrusive signatures

Here, Q= $\{q_0, q_1, q_2,\ldots.., q_{15}\}$

$\sum= \{0, 1\}$

$q_{0=}$ the initial state

F= $\{q_{10}, q_{14}\}$, dead state= q_{15} such that

$\delta(q_{15},0)=q_{15}$ and $\delta(q_{15},1)=q_{15}$

Thus the automaton reaches on the final state on input of the intrusive signature i.e. on states q_{10} or q_{14}. The presented automata checks more than one virus signatures (more specifically five signatures). All other signature strings reach to dead state for e.g. strings not starting or not ending with desired input value and so on.

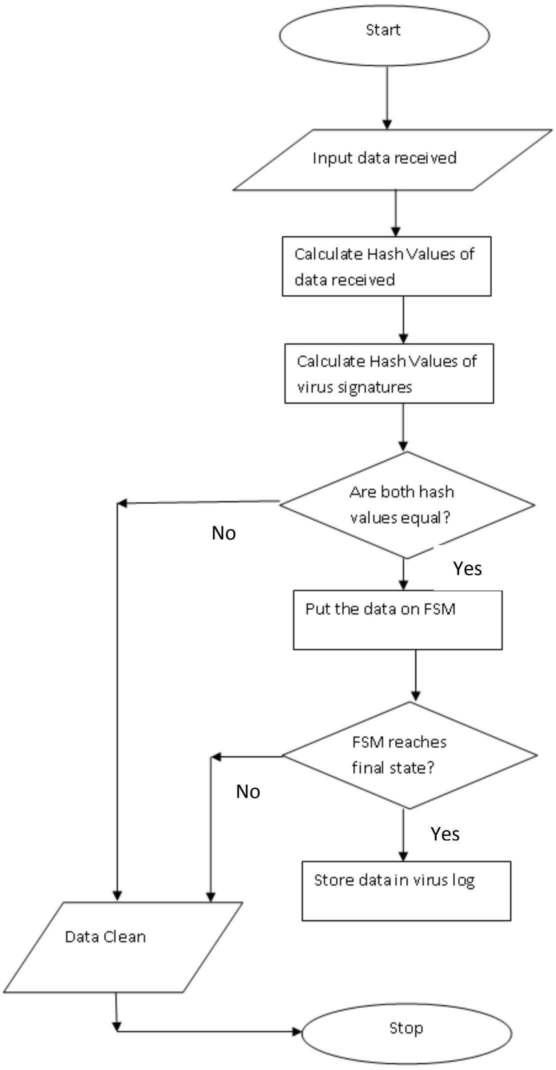

Fig2.Depicting System Flowchart

4. Conclusion

Preventing any network from intruders and making it free from malicious contents is of prime concern of the network security analysts. Intrusion detection along with intrusion prevention mechanism support lets any network secure from internal as well as external threats. Introduction of finite deterministic model into security areas like IDS is an innovative approach which can further be enhanced to meet more complex challenges. One of the benefits of using finite state model in our scheme is the usage of single model for same length digital signatures rather using separate models of same length finite automata. Also, by using Mid-Square method for hashing, the result is not dominated by the distribution of the bottom digit or the top digit of the original key value. The only limitation that exists in the presented system is in terms of non functionality of the system against such signatures which are not defined previously in the training database. Overall, the proposed paper presents an innovative approach in tackling intrusion attempts of a network and efficiently preventing the network from malfunctioning.

References

[1] James P. Anderson Co.; Computer security Threat monitoring and surveillance, Revised April, 1980

[2] Mohammadreza Ektefa : Intrusion Detection Using Data Mining Techniques

[3] Kingsly Leung, Christopher Leckie: Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters. In: 28th Australasian Computer Science Conference, The University of Newcastle, Australia, January 2005

[4] S. Hossain, S. M. Bridges, and R. B. Vaughn: Adaptive Intrusion Detection with Data Mining. In Proc. of IEEE Int'l Conf. on Systems, Man and Cybernetics, pp. 3097-3103, 2003

[5] Karen Scarfone & Peter Mell: A Guide to Intrusion Detection and Prevention Systems NIST Special Publication 800-94

[6] Hashing Tutorial: research.cs.vt.edu/AVresearch/