A Survey on Intrusion Detection System using Deep Packet Inspection for Regular Expression Matching

S.Priti1, S.Sumathi2, C.Amuthavalli3

1Assistant Professor, Department of CSE, RVS College of Engineering and Technology, Coimbatore, Tamil Nadu, India
2Associate Professor, Department of EEE, PSG College of Technology, Coimbatore, Tamil Nadu, India
3Associate Professor, Department of EEE, PSG College of Technology, Coimbatore, Tamil Nadu, India

Abstract: Intrusion Detection is the act of detecting unwanted traffic/access on a network. It plays the central concept in overall network and computer security architecture. Intrusion Detection System (IDS) is installed software or a physical appliance that monitors network traffic and network resources which detects unwanted activities such as illegal and malicious traffic, misuse of network resources etc. Deep Packet Inspection (DPI) is most widely used in IDS, DPI analyses all data present in the packet and uses regular expression matching as a core operator. The complex string patterns such as attack signatures in DPI are flexibly represented using regular expression. In this paper, the basics of IDS, its components, its types, various network attacks, levels of packet inspection, implementation of DPI and various methods of DPI, features of regular expression and the working principle of regular expression with DPI are discussed. Thus the intrusion detection engine using deep packet inspection for regular expression matching greatly improves the speed of the intrusion detection and can also meet the performance requirements of the network.

Keywords: Intrusion Detection System, Deep Packet Inspection, Regular Expression, Network attacks, Deterministic Finite Automata, Non Deterministic Finite Automata

1. INTRODUCTION

In recent year, usage of internet is increased in all the fields. As the usage of internet is increasing day by day, the security of the network is becoming essential in order to obtain security, integrity and confidentiality of a resource. Network security problems can vary widely and can affect the various requirements of security such as authentication, integrity, authorization, and availability. Intrusion Detection System plays an important role to keep our network secure. Intrusion detection is a process of monitoring and analyzing the data and events that occurs in a network in order to detect attacks, vulnerability and other security problems. An intrusion detection system (IDS) [4] provides a layer of defense that monitors the traffic in the network and alerts system administrators when potential hostile traffic is detected. There are a number of challenges that an intrusion detection system should face. Firstly, it must reliably detect the malicious behaviors in a network and must perform efficiently to cope with the large amount of network traffic. IDS can be a piece of installed software or a physical device. IDS tools must be used to store the detected event in a log to be reviewed at a later stage or must combine events with other data to make decisions regarding policies or damage control.

The DPI is a core component for many systems plugged in the network including proxies, packet filters, sniffers, IDS, and IPS. The components of the network use DPI as an essential inspector where it is applied in different layers of the OSI model. In the past, DPI was often accomplished by string matching, i.e., to find the strings in a set of predefined strings that match the payload of a packet. Now, DPI is typically achieved by regular expression matching, i.e., finding which regular expression in a set of predefined regular expressions match the packet payload, because regular expressions are fundamentally more expressive, efficient, and flexible for specifying attack or malware signatures. Regular expressions are used instead of simple string patterns because regular expressions are fundamentally more expressive and thus are able to describe a wider variety of attack signatures. Most open source and commercial Network Intrusion Detection System (NIDS)/Network Intrusion Prevention System (NIPS) such as Snort [22], Bro [23], and L7 Filter
2. INTRUSION DETECTION SYSTEM

Intrusion Detection System is defined as the software or hardware product, [4] which focuses and identifies probable incidents caused by attackers, monitors information about those intrusions, terminates them, and generates a report for security administrators [5] in real-time environment. So, Intrusion Detection System can be considered as a security operation that complements protection, e.g., firewalls [1]. It also helps to provide security and prevention against various intrusions caused by the attackers.

2.1 EVOLUTION OF IDS

For about two decades, intrusion detection has been an effective field of research [2]. In 1980, James Anderson published a manuscript, “Computer Security Threat Monitoring and surveillance”, is one of the earliest manuscripts in the field [1], [2]. Between 1983 and 1986, Dorothy Denning and Peter Neumann started working on the government project related to IDS development; they researched and proposed the first model of real-time IDS [1], [3]. The model was named as the “Intrusion Detection Expert System (IDES)”. First IDS ever developed was host based, because they are used to analyze the system log i.e., for analyzing the operating system log by evaluating the signatures with the small set of pattern or model. This leads to the growth of various IDS, until now the security faults were increased and the system performance was degrading owing to these security faults. This situation leads to qualified development of IDS at the earliest. Later IDS gained all the protocol awareness, used to analyze the packet, packet structures etc., which predicts the known packet traffic defined to be malicious. Currently recent IDS can predict the various attacks or types of intrusion through the network because the recent development in IDS leads to host communication, in built security features, protocol awareness, packet examining etc.

2.2 IDS COMPONENTS

Typical components of IDS and their functionalities [6] are discussed in the Fig.1.

- **Sensor/Agent**: Monitors and analyzes network activity. The term sensor is used for IDS that monitor networks, including network-based, wireless, and network behavior analysis technologies. The term agent is used for host-based IDS technologies.
- **Database Server**: Used as a repository for event information recorded by the sensors or agents processed by the management server.
- **Analyzer**: Centralized device that receives and analyzes the event information from the sensors/agents. It identifies events that the sensors/agents cannot. It sends alert to the manager. The purpose of the analyzer is to find out whether
the flow of events provided by the sensor contains features of malicious activity.

- **Manager:** The manager manages the various events and reacts with the operator about the event information that comes from the analyzer.

- **Console:** Provides an interface for the users and administrators. Console software is typically installed onto standard computers providing both administration and monitoring capabilities. IDS are differentiated by the types of events that they can recognize and the methodologies that they use to identify incidents.

### 2.3 FUNCTIONS OF IDS

The various functionalities of Intrusion Detection System include:

- **Recording Information:** Event information is usually recorded locally, and might also be sent to separate a system which includes centralized logging servers, security information and event management solutions, and enterprise management systems.

- **Notifying Security Administrators:** Alerts or alarms occur when any of the following like-emails, web pages, messages on the IDS user interface, Simple Network Management Protocol (SNMP) traps [10], syslog messages, and user-defined programs, are detected by the system. A simple notification message includes basic information concerning an event. The role of administrator is to access the IDS Console for additional information in order to neutralize them.

- **Producing Reports:** It represents summarized reports of the monitored events and actions taken by the administrator depending on the details of the particular events.

### 2.4 TYPES OF IDS

Several types of IDS technologies exist because of the variety of network configurations. Each type has advantages and limitations in detection, configuration, and cost. The types of IDS technologies are differentiated primarily by the types of events that are monitored and the ways in which they are organized.

#### 2.4.1 Host based Intrusion Detection System

It is used to monitor the characteristics of a single host and the events that occur in the host and also monitors and collects data on the host computer. These data are analyzed locally or aggregated to another computer for analytical operations. It makes use of two main approaches namely Anomaly Detection and Misuse Detection. Anomaly Detection refers to intrusions that are detected based on anomalous behavior and resources of the computer. Anomaly detection assumes that misuse or intrusions are highly associated to abnormal action obsessed either by a user or a system. Misuse detection techniques attempts to model attacks on a system as a specific pattern, and then systematically scans the system for occurrences of these patterns. Examples of the types of characteristics a host-based IDS monitor are network traffic, logs of the system, running processes, activity of the application, file access and modification, and changes in system configuration.

#### 2.4.2 Network Based Intrusion Detection system

Network Based IDS, which is named as NIDS, monitors and collects data on the network. It monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify suspicious activity. This system is potentially very good at detecting unauthorized users even before gaining access to the computer. Network based IDS observe the whole network traffic and generate the respective output of the analysis. NIDS are popular for its efficiency and robustness to detect and prevent attacks over the network. An example of a NIDS is SNORT.

#### 2.4.3 Signature-Based IDS

A signature based IDS monitor’s packets in the network and compares with preconfigured and predetermined attack patterns known as signatures. When a new attack is recognized the experts or programs identify typical patterns in the attacks, which are made into signature. Since this process takes time, there will be a lag between the new threat revealed and signature that are applied in IDS for detecting the threat. During this lag time the IDS will be unable to identify the threat. To reduce further lag, security software using such signatures should be updated as frequently as feasible.

#### 2.4.4 Distributed Intrusion Detection System

In this type of IDS there is no central processing IDS acting but instead at different host there are different IDS engaged working inaccessible to others however coordinates with each others. This technique decreases detection time by decreasing processing load at one central engine. Another
definition of Distributed IDS is that it consists of large number of intrusion detection systems distributed over large network, which communicate with each other or the central server which is responsible for monitoring the network by analyzing incidents and generating the related patterns.

3. DEEP PACKET INSPECTION

Deep Packet Inspection (DPI) is a technology that allows the network owner to analyze internet traffic, transmitted through the network, in real-time and to differentiate traffic on the basis of their payload. Originally the Internet protocols need the network routers to scan only the header of an Internet Protocol (IP) packet. The packet header includes the origin and destination address and other information about to moving the packet across the network. The “payload” or content of the packet that contains the text, images, files or applications send by the user, was not considered to be a concern of the network operator. DPI enables network operators to scan the payload of IP packets as well as the header.

3.1 LEVELS OF INSPECTION

The functions provided by DPI technology are accessible before to inadequate degree depending on the level of packet analysis. Packet inspection technologies that are used in networking environments can be classified into three classes. These three classes are shallow, medium, and deep packet inspection. Fig.2 [8] provides a visual representation of the depth of inspection each of these technologies allows for.

3.1.1 Shallow Packet Inspection

Shallow packet inspection (SPI) examines the headers of the packets information that are placed at the beginning of a block of data which includes the sender and recipient's IP addresses. This kind of packet inspection allows the communications to remain almost unspecified because the content of the packets is not experimented, and the information in the header is used only to route the packet.

3.1.2 Medium Packet Inspection

Medium Packet Inspection (MPI) refers to application proxies or devices that are positioned between end-users computers and ISP/Internet gateways. These proxies examine the packet header information in opposition to their loaded parse-list. When a packet enters the proxy it is analyzed beside a parse-list that system administrators can easily update. A parse-list allows or disallows specific packet-types depending on their data format types and associated location on the internet, rather than on their IP address alone.

3.1.3 Deep Packet Inspection

Deep Packet Inspection (DPI) technologies are proposed to permit network operators specifically to identify the source and content of each data packet passes through the networking hubs. Whereas MPI devices has very limited application responsiveness, DPI devices are possible to look inside all traffic from a specific IP address, choose out the HTTP traffic, then drill even further down to confine traffic headed between a specific mail server, and then reassembles e-mails as they are typed out by the user. DPI devices are deliberated to determine the programs that generate packets, in real time, for hundreds of thousands of transactions each second.

3.2 METHODS OF DPI

The concept of DPI inspects both the header and the payload content and they can be used for traffic classification or IDS. In Fig. 3 the various popular methods of DPI [19] is specified.
The port based approach [13] is the classical method used for protocol identification. It is done by checking the port fields in TCP or UDP headers and the well known port numbers are assigned for protocols. Statistical Analysis approach [14] is mainly focused on traffic classification which is payload independent. It collects information such as port numbers, packet length, timestamp, etc. to characterize traffic. Finsterbush et al. [12] mainly focused protocol decoding method to characterize traffic classification. It is a light weight pattern matching which recognizes protocols. It recognizes protocols by characteristics of protocol headers such as magic numbers, session identifiers, etc. and by the behavior of protocols.

Heuristic based approach [15] is concentrated on exact string matching. During the matching, a window with length of m covers the characters waiting to be inspected, and the matching result indicates the next position and the window should slide to. It is used to identify the appearance of patterns prefix and to find next position. In hashing based approach [16], for each string pattern of length m, a hash value is computed for the payload and the patterns are matched. This method quickly judges whether a substring in the payload match a set of patterns. Whereas a filtering based approach easily excludes the characters that do not match a pattern.

Automaton based approach [11], [17-18] are the most commonly used method for pattern matching which extensively uses pattern matching or regular expression matching. A finite state automaton is generated for the patterns and the automaton consists of an initial state and some accept state that indicates the matching of patterns, and other intermediate states represent partial matching situations. The matching process starts from the initial state, each time a payload character is sent to the automaton for state migration. If a final state is visited during the process, the corresponding pattern(s) match is found.

4. NETWORK ATTACKS

Any kind of malicious [7] activity that tries to collect, infest, decline, debase, or impair information to the system resources or the data itself. An attack can be active or passive.

4.1 PASSIVE ATTACK

Passive attack tries to learn the use of information from the system but does not clash the system resources.

- **Wiretapping**: Third party monitors the covert information from a telephone line or network. The secret connection will be real electrical tape of the telephone line.
- **Release of message content**: Telephone conversation/ Email messages/ Transferred file contain some secret data. Attacker monitors the content of these secret transmissions.
- **Traffic Analysis**: Attacker analyzes the traffic, determine the location, identify communication hosts, and observe frequency and length of messages. All incoming & outgoing traffic of network are analyzed but not altered.

4.2 ACTIVE ATTACK

The motto of the active attacker is to change the information in the network.

- **Denial of Service**: In a network the host can get the same information from the same server for multiple times. This causes overloading of data. By using this limitation the attacker tries to get that server for multiple times. Resulting which the services to the genuine host will be blocked. In denial of services attack, malicious node sends the message to the node and consumes the bandwidth of the network. The main aim of the malicious node is to be busy over the network node. If a message from unauthenticated node comes, then receiver will not receive that message because he is busy and beginner has to wait for the receiver response.
- **Spoofing**: The identity of the malicious node is miss-presented, so that the sender changes the topology. One program successfully pretends as another by sending wrong data. For example, DNS Spoofing.
- **Man-in-the-middle**: The attacker continuously watches the communication between two parties. The attackers make independent connection between them and relay the messages.
- **ARP Poisoning**: The attacker sends spoofed ARP messages onto the Local Area Network. Spoofing may allow an attacker to modify or stop all traffic.
- **Buffer Overflow**: While writing data to a buffer replaces adjacent memory location. This is a special
Cyber Attack: Any type of aggressive operation utilized by individual that target the computer data, infrastructure, network information.

Phishing attack: It obtain sensational information such as user name, password and credit card details. The attacker tries to get these details and modifies these messages.

Modification: When malicious node performs some modification in the routing table, the sender sends the message through the long route. This attack cause communication delay occurred between sender and receiver.

Wormhole: This attack is also called the tunnelling attack. In this attack an attacker gets a packet at one point and tunnels it to another malicious node in the network. By doing this the shortest path in the network is found.

Fabrication: A malicious node generates the false routing message. This means it generates the incorrect information about the route between devices.

Sinkhole: Sinkhole is a service attack that prevents the base station from attaining whole and correct information. In this attack, a node tries to attract the data to it from all neighboring node. Selective modification, forwarding or dropping of data can be done by using this attack.

Sybil: This attack is related to the multiple copies of malicious nodes. The Sybil attack happens due to malicious node share its secret key with other malicious nodes. Thus the number of malicious node is improved in the network and the probability of the attack also increases. If we use the multipath routing, then the possibility of selecting a path of the malicious node will be increased in the network.

Then, for each character in the input, the FA makes a transition to the next state, which is determined by the previous state and the current input character. If the resulting state is distinctive, the Finite Automata is called as a deterministic finite automaton; otherwise, it is called as a nondeterministic finite automaton. For the exchange between the performance and usage of the resources, NFA and DFA are represented as two extreme cases. DFA has constant time complexity since it guarantees only one state transition for each character by definition. The constant time complexity for each character allows DFA to achieve high-speed regular expression detection, which makes DFA the preferred approach. This is because for software solutions DFA are serially executed rapidly on service CPUs. NFA, on the other hand, requires massive parallelism, making it harder to implement in software. NFA allows multiple simultaneous state transitions that lead to a higher time complexity. The cost of the high speed DFA is its prohibitively large memory requirement.

5.1 FEATURES OF REGULAR EXPRESSION

A regular expression illustrates a set of strings without specifying them explicitly. Table 1 lists out the general features of regular expression patterns that are used in packet payload scanning.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>^</td>
<td>Pattern matched at the start of the input</td>
<td>^CD means the input begins with CD. A pattern without &quot;^&quot; e.g., CD can be matched anywhere in the input.</td>
</tr>
<tr>
<td>[ ]</td>
<td>OR relationship</td>
<td>X</td>
</tr>
<tr>
<td>?</td>
<td>A quantifier that denotes one or less</td>
<td>X? denotes X or an empty string.</td>
</tr>
<tr>
<td>*</td>
<td>A quantifier that denotes zero or more</td>
<td>X* means an arbitrary number of Xs.</td>
</tr>
<tr>
<td>{ }</td>
<td>Repeat</td>
<td>X[100] denotes 100 Xs.</td>
</tr>
<tr>
<td>[ ]</td>
<td>A class of characters</td>
<td>[abc] denotes a letter a, b or c.</td>
</tr>
<tr>
<td>[^]</td>
<td>Anything but</td>
<td>[^\n] denotes any character except \n.</td>
</tr>
</tbody>
</table>

Table 1: Features of Regular Expressions [9].
5.2 NON DETERMINISTIC FINITE AUTOMATA (NFA)

A nondeterministic finite state automaton is a machine model with simple memory ability. A non-deterministic finite automaton NFA [24] M is a five tuple, and its mathematical definition is $M = (S, \Sigma, F, s_0, A)$. These five elements are explained as follows:

- $S$ is a finite set of states, and its every element is called as a state;
- $\Sigma$ is a finite set of input alphabets, each of which is called as an input symbol;
- $F$ is a transition function;
- $s_0 \in S$ is a non-empty start state;
- $A \subseteq S$, it is an accept state set, and the accept state can be accepted as the state or the end state.

5.3 DETERMINISTIC FINITE AUTOMATA (DFA)

A deterministic finite automaton DFA [25] M is a five tuple, and its mathematical definition is $M = (S, \Sigma, F, s_0, A)$. These five elements are represented as follows:

- $S$ is a finite set of states, and its every element called a state;
- $\Sigma$ is a finite set of input symbols, each of which is called an input symbol;
- $F$ is a conversion function, which represents the current state of $S_i$, when the input character is $a$, it will be converted to the next state $S_j$, and we call $S_j$ a successor state of $S_i$;
- $s_0 \in S$, it is the only one initial state;
- $A \subseteq S$, it is a final state set, and the final state can be accepted as the state or the end state.

The regular expression is equal to NFA, that is, each regular expression can be modeled as a NFA which matches the same string and vice versa. There are many kinds of methods to convert regular expression to the NFA however the most famous method is the construction algorithm that was proposed by Thompson in 1968 [21].

5.4 REGULAR EXPRESSION AND DEEP PACKET INSPECTION

Regular expressions are replaced by precise string patterns as the language of pattern matching in packet scanning applications. The widespread use of regular expression is because of its supreme authority in expression and it is flexible to describe useful patterns. For example, in the Linux Application Protocol Classifier L7 – Filter [20], all protocol identifiers are expressed as regular expressions. For example, let’s take a regular expression from the Linux L7 - filter that detects Yahoo traffic: “^(ymsg | ypns | yhoo ).?\.?\.\?\.\.\?\?\?\?\?[lwt]*\xc0\x80\x80\". This regular expression matches any packet payload that begins with ymsg, ypns, or yhoo, followed by seven or fewer arbitrary characters, and then a letter l, w or t, and some arbitrary characters, and finally the ASCII letters c0 and 80 in the hexadecimal form.

In network applications, regular expression matching is mostly used for application protocol identification and NIDS, and regular expression patterns are represented by the unique characteristic of an application-level protocol, a virus, a spam or a malware. Due to its significant expressiveness and capability of flexible description, regular expression has been extensively used in several open source DPI applications and commercial DPI engines, and the development is still growing.

6. CONCLUSION

In this paper a comprehensive survey is done on the various types of Intrusion Detection System and its components. The levels of packet inspection and the various popular methods of DPI are discussed. The automaton based approach plays an important method in pattern matching for DPI. In this method, regular expression matching is the main tool for many DPI applications where the packet payloads are inspected against a set of patterns. The features of the regular expression and how it can be applied with deep packet inspection are discussed. The various network attacks that occur in the network are explained. Because of powerful and flexible expressive ability of regular expression, regular expression matching can be used as a major tool for pattern matching in DPI which identifies the various attacks in the packet payload. Thus by using regular expression matching in intrusion detection engine the performance requirement of the network improves and there will be a rapid improvement in the intrusion detection speed.

REFERENCES

[1] Asmaa Shaker Ashoor (Department computer science, Pune University) Prof. Sharad Gore (Head


