Comparative Network Intrusion Detection Technique DPI-AD for Regular Expression Detection in Wireless Ad-Hoc Network

Mr. Girish M. Wandhare[1] Prof. S. N. Gujar[2] Dr. V. M. Thakare[3]
[1]M. E Student
[2]Department of Information Technology
[1,2]SKNCOE, Pune Maharashtra, India [3]S.G.B. Amravati University, Amravati Maharashtra, India

Abstract— DPI-AD is a deep packet inspection technique for regular expression detection in wireless network. This technique design in the base of LaFA technique, which is use to detect regular expression of attacks on wireless network. DPI-AD used DDOS attack filter for detection of DOS attack. Regular expression detection module used to detect regular expression pattern for various attacks. LaFA uses pattern matching in simple detection and complex detection module which require more time for detection of attack pattern for regular expression attacks. The detection technique in LaFA detect cross site scripting (XSS) attack, where the SQL injection attack cannot be blocked. The DPI-AD recovers all its limitations and additionally it gives better detection results. For the testing purpose of this techniques Sql injection attack, XSS attack and DOS attack tests are done on DPI-AD. Other Regular expression techniques also discuss with compare to DPI-AD.

Key words: Deep Packet Inspection in Ad-hoc wireless network(DPI-AD) Deep Packet Inspection(DPI); Regular Expression(RegExp); Deterministic Finite Automata(DFA); LaFA; StriFA; CompactDFA; Tcam; DFA/EC; Snort; Bro.,

I. INTRODUCTION

Deep Packet Inspection (DPI) technology allows the network administrators to analyze internet traffic, through the entire network, in real-time and to separate them according to their payload [3]. Old packet inspection algorithms have been limited to match packets to a set of strings. Newer DPI systems, like Snort [11] and Bro [10] uses rule-sets containing regular expressions, systems like these are costlier, efficient, and compact in identifying attack signatures [4].

Regular expressions (RegExes) uses by the most of the applications to represent complex string patterns in various applications, like network intrusion detection and prevention systems (NIDPSs), DNA multiple sequence alignment and Compilers [1]. Network intrusion detection is the formation of scan the events occurring in network or a computer system and analyzing them for chances of possible incidents, which are imminent threats of trespass of computer security and standard security policies. Intrusion detection and prevention systems (IDPS) are initially focused on specifying possible incidents in data packet and logging information related to these incidents, try to stop and reporting them to network security administrators [3].

Network intrusion detection system techniques like Bro [10] and Snort [11] use RegExes to represent signatures of attack pattern. Regular expressions serve as complex string pattern similar to attack signatures in most of the applications. Current regular expression detection system is incapable of supporting large RegEx sets which are expected in future time with high speed requirement [1]. LaFA technology based on following three observations, first is RegExes contains a variety of components like character classes or repetitions of characters, Second the order of these components in a RegEx is conserve in the state machine detection of RegEx . Third is most of the RegExes share similar components with each other [1]. The LaFA needs small amount of memory because of these three contributions: 1) It provides optimized and specialized detection modules to increase utilization of resources, 2) The sequence of RegEx detection reordering to minimize the number of concurrent operations systematically; 3) It shares states among automata for various RegExes to minimize the requirements of resource.

The TCAM is the first hardware-dependent RE matching technology that uses ternary content addressable memory (TCAM), The TCAM is widely deployed in recent networking devices for tasks like packet separation. The StriFA [6] technology introduce the stride finite automata, it is a unique family of finite automata. It is used to speedup both string matching and regular expression matching systems. The Compact DFA [6] proposed technology to compress DFAs by analyzing that the name used by traditional DFA encoding is meaningless in the method. This degree of freedom and states of encode in the way that all transitions to a particular state are denoted by a prefix which defines a set of current states. The Compact DFA methodology applies to a huge class of automata, which can be differentiating by simple properties. In the TCAM [2] technology, the throughput of compact DFA reaches up to 10 Gb/s with low power consumption requirement. This technique uses Aho–Corasick (AC) algorithm, that uses a deterministic finite automaton (DFA) to the pattern set represent [6].

Extended Character set DFA [3] concentrate on minimizing the requirement of memory storage of DFA, and it can be differentiating into the following categories: minimizing the number of states, reducing the number of transitions requirements, reduction in the bits encoding the transitions requirements, and minimizes the character-set. Unfortunately, these approaches compress DFAs at the price of increased access of main memory. This methodology propose a novel solution, known as deterministic finite automata with extended character-set (DFA/EC), it can continuously reduces the number of states via doubling the size of the character-set. This technology solution requires only a single main memory access for every byte in the traffic payload on the network [3].

This implementation consists of improved RegEx detection technique which will have higher throughput than any other network intrusion detection techniques. We will minimize memory requirements and resource usage by network detection system. The project will be used for regular expression detection of attack signature pattern matching in wireless network. Modules of the projects are survey existing

techniques, implement DOS attack filter, built RegEx detection technique which will work on wireless network.

II. PAPER ORGANIZATION

The rest of the paper organized as follows. Literature survey for this comparison given in Section 3. Here we describe Detail literature survey of DPI and technologies. Section 4 describes the existing system problems. Section 5 contains system implementation details. Mathematical model for problem statement describes in section 6. Result and comparison details showed in section 7. Section 8 includes conclusion and future work.

III. LITERATURE SURVEY AND ELABORATION & SYNTHESIS

A. Deep Packet Inspection

The Deep Packet Inspection (DPI) is a technology which allows the network administrator to examine internet traffic, on the network in real-time environment and to vary them depending on their payload. This needs to be done on real time basis at the efficient speeds which cannot be implemented by software running processors or switches. It has only become possible in the recent few years by advances in computer engineering technologies and in pattern matching algorithms [2].

The Internet protocols need the network routers to examine only the header of an Internet Protocol (IP) packet. The header packet includes the source and destination IP address and other information regarding to transmission the packet through the internet network. The packet "payload" or content contains the text, files, images or transmitted applications by the user, was not concentrate to be a concern of the network operator or network owner. DPI enables network operators to analyze the IP packets payload as well as the header of packet. Figure 3.1 [8] shows the domain of packet analysis required in internet protocols and in DPI [8].

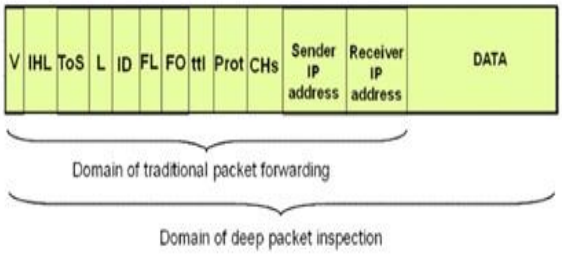


Fig. 3.1: Domain of Deep Packet Inspection [8]

DPI technology uses regular expressions to define patterns of interest in data streams on the network. The equipment is programmed to take decisions like how to handle the packet or a stream of packets depends on the identification of a regular expression or pattern in the payload of packet. This technique enables networks to differentiate and control network traffic on the basis of the applications, content, and subscribers [8].

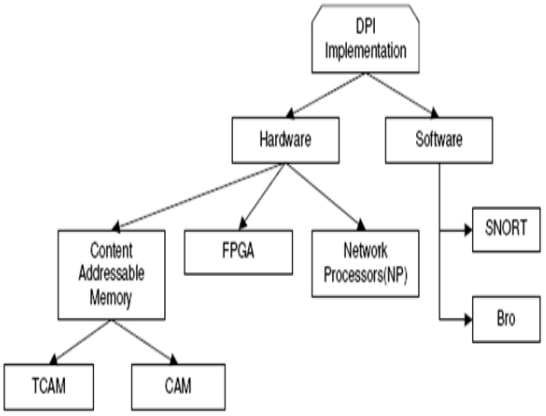


Fig. 3.2: DPI Implementation [7]

1) Regular Expression:

Regular expression (RE) typically use by deep packet inspection to matching as a core operator. DPI analyze whether a packet's payload matches any of a predefined regular expressions set. Regular expressions are fundamentally more efficient, expressive, and flexible in identifying attack signatures. RE matching algorithms are either software based or field-programmable gate array (FPGA) based [1].

Regular expressions contain a number of different components like repetitions or character classes [1]. Because of this variety, it is difficult to specify a method which is efficient for detection of all these various components of a RegEx. Many RegExes share components which is similar to each other. In the traditional FA, for detection of a component in a RegEx, a small state machine is used. This state machine is duplicated thus the similar components may appear number of times in various regular expression. Most of the time, regular expression shares these components doesnot appears at the same time in the input. It results in the repetition of the same state machine for various regular expression causes redundancy and limits the regular expression detection scalability of the system. Following figure shows the example illustrating the transformation from a regular expression set R into the corresponding LaFA technique [1].

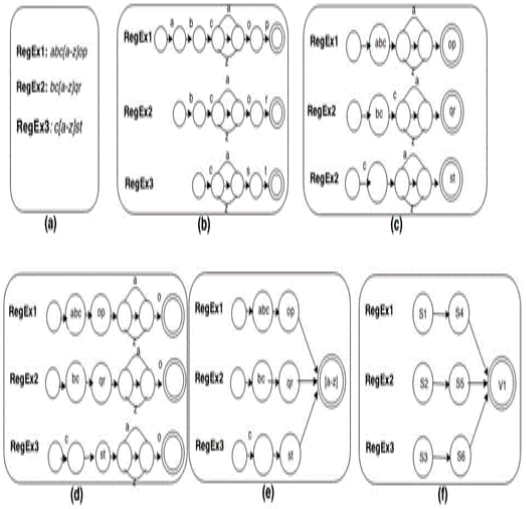


Fig. 3.3: Example illustrating the transformation from a RegEx set R into the corresponding LaFA. (a) RegEx set. (b) NFA corresponding to. (c) Separation of simple strings. (d) Reordering of the detection sequence. (e) Sharing of

complex detection modules. (f) LaFA representation of the RegExes [1].

As regular expressions achieve widespread acceptance for scanning of packet content, matching regular expression over the payload of packet similar with the line-speed processing of packet header is important. This requirement cannot be met in scanning of many existing payload implementations unfortunately. As example, when all 70 protocol filters are work in the Linux L7-filter [1], research found that the system throughput reduced to less than 10Mbps, that is less than current speeds of LAN . Ferther more, over 90% of the CPU time is spent in matching regular expression, saving little time for other intrusion detection and functions monitoring. On the other hand, many schemes for fast matching string have been recently developed in intrusion detection systems, they concentrate on only explicit string patterns and cannot be extended to fast regular expression matching easily [9].

Regular expression describes a set of strings by without enumerating them explicitly. Table 3.1 lists the common features of regular expression patterns used in payload scanning of packet. For example, suppose a regular expression from the Linux L7-filter for identifying Yahoo traffic: "^(ymsg|ypns|yhoo).?.?.?.?.?.?[lwt].*\xc0\x80". This pattern compare with any packet payload which starts with ypns, yhoo, or ymsg, followed by seven or less arbitrary characters, then a letter l, t or w, and some arbitrary string characters, and finally the ASCII letters c0 and 80 in the hexadecimal form[9].

Syntax	Meaning	Example
^	At the start of the input pattern to be matched	^AB denotes the input starts with AB. A pattern without '^', e.g., AB, can be matched in the input anywhere.
I	OR relationship	*A*|*B* means *A* or *B*
.	A single character wildcard	
?	A quantifier means one or less	*A*? denotes *A*, or an empty sting.
*	A quantifier means zero or more	*A** means an arbitrary number of *A*s.
{}	Repeat	*A*{100} denotes 100 *A*s.
[]	A class of characters	[*lwt*] denotes a letter *l*, *w*, or *t*.
[^]	Anything but not n	[^\n] denotes any character except \n.

Table 1: Features of Regular Expressions [9].

2) Deterministic Finite Automata (DFA):

The DFA contains a finite set of input symbols (denoted as P), a finite set of states, and a function of transition to travel from one state to the other symbol as @. Compare to NFA, DFA has one active state only at any given time [4][9].

For packet payload inspection to different protocols packet the regular expression is required, which implemented a limited DPI system to handle with all packets structures on the network. This limitation gives result of state-of-art systems which have been introduced to take place of the string sets of intrusion signature with more express regular expression systems. Hence, there are number of content inspection engines that have partially or completely migrated to regular expression including in Bro [10], Snort [11], and Cisco systems'.

As experimental results, DFA of regexp which contains hundreds of pattern starts to tens of thousands of states that mean consumption memory in hundreds of megabytes. For a solution of one of the similar problems of hardware based DPI solutions is the memory access due to the accesses of memory for the off chip memory contents are proportional to the number of the packet bytes [9].

B. Related Work:

1) Lafa [1]:

LaFA is a novel method for regular expression detection which resolves issues of scalability for the current RegEx detection paradigm. It is finite automata implemented for scalable RegEx detection purpose. It is used for representing a RegExes set (R={r1;r2;r3;…}) that can also be known as a RegEx database. LaFA RegEx detection system can be queried with an input like a network packet. This system will result a match along with a matching RegExes list if input contains one or more of the RegExes stream in it. Otherwise, it gives no match in returned. This technology gives facility to decrease in memory requirements and complexity of detection.

LaFA architecture shown in following figure 3.4 consists of two blocks, these are the detection block and other is the correlation block. Component detection is done by the Detection Block for the input string.

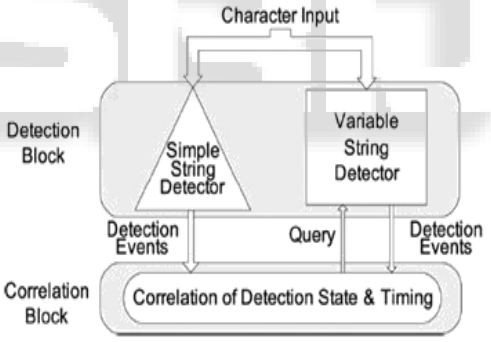


Fig. 3.4: LaFA Architecture [1].

The variable string detector contains highly optimized modules for variable string detection. Status of RegEx track by the correlation block. This block examines the sequence of components of the input string. The detection block detectors communicate their findings with the correlation block by sending detection events. Every detection event contains a unique ID for the detected component and its location in the input packet.

2) Small TCAM [2]:

This technology is the first ternary content addressable memory (TCAM)-based on RE matching solution. It uses a TCAM and its supporting SRAM to encode the DFA transitions built from an RE set where a TCAM entry might encode number of transitions of DFA. There are three key reasons why TCAM-based RE matching works efficient given in this research: first is a small TCAM is able to encoding a large DFA with carefully developed algorithms. It facilitate high-speed regular expression matching because of TCAMs are essentially high-performance parallel lookup

systems These TCAMs are off-the-shelf chips that are deployed in latest networking devices; it should be simple to developed networking devices that built with the TCAM-based RE matching solution[2].

In this methodology, TCAM can store 25 K states of a DFA in a 0.5-Mb TCAM chip; most DFAs need at most one TCAM entry per DFA state. With the use of variable striding it gives a throughput of up to 18.6 Gb/s is possible.

3) Strifa [5]:

StriFA methodology has been implemented in software. This methodology evaluation based on different traces. The StriFA handles the problem of implementation of a variable-stride pattern matching engine. This technology can achieve a maximum RE matching speed less use of memory. It proposes stride finite automata (StriFA); which can examine a verity of number of characters at a time. StriFA is designed to immune to the memory blow-up and problems of byte alignment; Hence, It needs much low memory than the previous rematching schemes. StriDFA (Stride deterministic finite automaton) and StriNFA (stride nondeterministic finite automaton) are two basic forms of StriFA implementation.

The results of this methodology showed that this architecture can reach about 10-fold increment in speed of matching, with a less consumption memory compared to traditional NFA/DFA technologies, while capabilities of the same detection maintain

4) Compactdfa [6]:

CompactDFA gives advantage that it fits into IP-lookup solutions which are commercially available. These solutions may be used for performing fast pattern matching. Output of the CompactDFA scheme is a compressed rules set, such that there is only one rule per state in the set. In the compressed rules set, a state set may compare with multiple rules of pattern matching. The algorithm implemented in CompactDFA presents the intuition behind each of its three stages: The state grouping, the Construction of common Suffix Tree and the State and Rule Encoding.

In this technology evaluation is done only for the process of pattern matching. It uses two common pattern sets: The Snort and the ClamAV. This technology can result fast pattern matching up to 2Gb/s with less power consumption. CompactDFA can achieve fast pattern matching up to 2 Gb/s with less power consumption. Because of this small memory and power requirements, proposed architecture can built with multiple TCAM working in parallel. Every TCAM match the pattern on a different session and achieve a total throughput of 10 Gb/s.

5) A DFA With Extended Character-Set [4];

This technique implementation depends on general-purpose processors which are flexible to update and cost-effective. It provides a novel solution known as deterministic finite automata with extended character-set (DFA/EC) that can significantly reduces the number of states by doubling the character-set. This methodology solution requires only access of a single main memory for each traffic payload byte. It experiment with several rule-sets of Snort. This technology results show that, compared to DFAs, this algorithm are very compact and over four orders of magnitude less in the best cases.

DFA/EC is a general DFA model which incorporates a part of the DFA state into the input characters set. This algorithm gives an efficient implementation of the inspection program depends on DFA/EC model given in this methodology that results in a compact transition table and increased inspection speed. It proves that DFA/EC technology is similar to DFA technique. It evaluates DFA/ EC with related algorithms by use several Snort rule sets.

The DFA/EC algorithm selects some of the most regularly active states of NFA and incorporates them into the DFA character-set to build a slightly bigger extended character-set. This methodology have additional constraints that exclude some of the regular active NFA states from the set of complementary states to enable method of a single-bit encoding complementary states in the extended character set, and to used it for efficient DFA/EC implementation.

The requirement of total minimum memory for the transition tables in terms of the number of bits and is the product of the transitions number and the bits count required to encode every measured transition. It results; the memory of DFA/EC bandwidth can even less than DFA in web-misc-28 and rule-sets exploit-19.

6) Dos Attack:

A Distributed Denial of Service (DDoS) attack is an attempt to make an online service unavailable by overwhelming it with traffic from multiple sources. They target a wide variety of important resources, from banks to news websites, and present a major challenge to making sure people can publish and access important information.

Attackers build networks of infected computers, known as 'botnets', by spreading malicious software through emails, websites and social media. Once infected, these machines can be controlled remotely, without their owners' knowledge, and used like an army to launch an attack against any target. Some botnets are millions of machines strong. Botnets can generate huge floods of traffic to overwhelm a target. These floods can be generated in multiple ways, such as sending more connection requests than a server can handle, or having computers send the victim huge amounts of random data to use up the target's bandwidth. Some attacks are so big they can max out a country's international cable capacity.

a) Difference Between DOS And DDOS Attack:

It is important to differentiate between Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. In a DoS attack, one computer and one internet connection is used to flood a server with packets, with the aim of overloading he targeted server's bandwidth and resources.

DDoS attack, uses many devices and multiple Internet connections, often distributed globally into what is referred to as a botnet. A DDoS attack is, therefore, much harder to deflect, simply because there is no single attacker to defend from, as the targeted resource will be flooded with requests from many hundreds and thousands of multiple sources.

7) XSS And SOL Injection Attack:

a) Xss Attack:

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

XSS attacks may be conducted without using <script></script> tags. Other tags will do exactly the same thing, for example:

```
<body onload=alert('test1')>
```

or other attributes like: onmouseover, onerror.

Onmouseove

```
<b onmouseover=alert('Wufff!')>click me!</b>
```

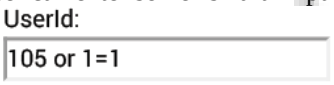
Onerror

```
<img src=http://url.to.file.which/not.exist
onerror=alert(document.cookie);>
```

b) Sql Injection Attack:

SQL injection is a technique where malicious users can inject SQL commands into an SQL statement, via web page input. Injected SQL commands can alter SQL statement and compromise the security of a web application.

Let's say that the original purpose of the code was to create an SQL statement to select a user with a given user id. If there is nothing to prevent a user from entering "wrong" input, the user can enter some "smart" input like this:



c) Server Result

```
SELECT * FROM Users WHERE UserId = 105 or 1=1
```

The SQL above is valid. It will return all rows from the table Users, since WHERE 1=1 is always true. The example above seems dangerous, What if the Users table contains names and passwords?

The SQL statement above is much the same as this:

```
SELECT UserId, Name, Password FROM Users WHERE
UserId = 105 or 1=1
```

A smart hacker might get access to all the user names and passwords in a database by simply inserting 105 or 1=1 into the input box.

IV. PROBLEM WITH EXISTING SYSTEMS

From the survey of above techniques, we found that, the approach describe in these techniques may require a large number of transitions for some cases, leading to an increase in the number of memory accesses per input byte. In addition, DFA construction is complex and requires significant resources[1]. There is very few network intrusion detection techniques discover in wireless networks.

CompactDFA technique used in architecture requires several TCAM working in parallel, Due to its small memory and power requirements. NBA technologies have some significant limitations. They are delayed in detecting attacks because of their data sources, especially when they rely on flow data from routers and other network devices[3]. DFA/EC does not combine with the existing transition compression and character-set compression techniques, and perform experiments with more rule-sets[4]. One of the problems for StriFA is how to choose an appropriate tag. Since in both the rules and the incoming traffic, the occurrence probabilities of different characters vary from each other, it is a problem to choose an appropriate tag from the rule set [5].

Following table 4.1 shows comparison of existing network intrusion detection techniques.

Intrusion Detection Techniques	Throughput
LaFA	34 Gb/s
CompactDFA	10 Gb/s
Small TCAM	18.6 Gb/s
StriDFA	26.5 Gb/s

Table 2: Comparison of deep packet intrusion techniques

A. Objectives of Problem:

Following are objectives of problem we define on the basis of above survey:

1) To improve network intrusion detection throughput with use of DPI techniques.
2) LaFA technique can be modifying for effective detection of evaluating RegExp on the network.
3) Higher throughput in network intrusion detection can be possible.
4) Techniques discuss above could have efficient performance in requirements of memory, detection speed of intrusion and detection of evaluating RegEx detection.
5) There are limited intrusion detection techniques which work on wireless network.

V. SYSTEM IMPLEMENTATION

A. Module Partitioning

This section we include the partitioning of project into different modules. All these modules are explained as follows:

1) Client and Server:

This module includes the registration and log in of client and server. It includes following classes.

- Log In : With this class client can login to the server web site i.e. Banking website
- Register: this class used to register new user for server site.
- Admin Login: Admin can log in and manage the data of website.
- Admin View Log: With this class admin can view all the block IP and type of attacks done and its date and time.
- Unblock: With this class admin can unblock the block IP.

2) DDoS Attack Filter:

- Request Count: This class count no. of requests per second (RPS) coming from client.
- RPS Limit: This class used to save the limit of the request per second any human user can make.
- Block IP: DDOS filter block the user IP if RPS exceeds the limit

3) DPI Module:

- Pattern Matching: This class used deep packet inspection pattern matching algorithm to detect the attack pattern.
- Block IP: If request packet pattern match to attack pattern saved in database then it block the client IP.

4) Database:

- In this project we use Apache Tomcat sever to use database named IDS.
- This database used to save attack pattern and request count per second and access by DDOS filter and DPI module to search for the attack pattern.
- Database used to store account data, user account id and password, list of blocked ip, type and date and time of attack when it occurs.

B. Low-Level Design:

Low-level designs of software system include:

- private classes, private methods, private attributes
- Algorithms.

Low-level design also provides an interface for all classes, public and private methods, including parameters, return values, exceptions thrown and types defined. It describes and justifies the choice of data structures, describes the major alternatives that are considered and why the choice is preferred that is opted.

In our project private classes such as block ip class, account information class only can be access by owner. Owner can login to the server using admin login and see the all information from the site. Admin also have privileges to alter the data saved in database such as unblock the user ip. Client can only access the information about his account, hr does not see any attack information or other user account data. If client try to make attack on the server it can block by dos filter and DPI module.

DOS filter compare the request count per second with the possible no. of the request any human user can make which is saved in database and make the decision to block the ip or not. DPI used to compare packet pattern with the attack pattern saved in database if attack pattern match then it blocks the user op who try to attack on server.

C. Implementation Setting:

We implement our project on Net beans IDE 7.3. Project is implemented in java language. We used SQLyog enterprises to use Apache Tomcat server. Client server model is used to run this project. To connect server and client we implement Ah-hoc wireless connection between them. We also test our project on internet by connecting internet connection. Data owner & authorized user can login through any machine.

VI. MATHEMATICAL MODEL OF PROBLEM STATEMENT

Mathematical Model for simple regular expression matching
Consider
R=regular expression
R= abc[a-z]op
P=pattern with which we match the input packet regular expression.
If
R=P (Regular expression pattern match)
i.e R(i)=P(i)
R(ii)=P(ii)
R(iii)=P(iii) . . . . .
Then the packet blocked
Else the packet forward to the server.
In our approach we divide regular expression in two division
S= set if simple variables in regular expression.
C= set of complex variables in regular expression.
Consider
R= adc[a-z]op
We divide this regular expression
S= abcop
C= [a-z]
If
S=P (Pattern Matching)
C=P (Pattern Matching)
Then the packet is block
Otherwise packet forwarded to the server.
Figure 6.1 shows RegEx components based on their detection complexity and depth.

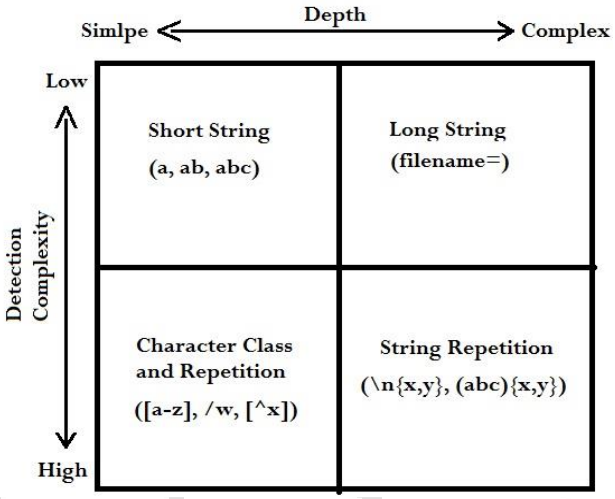


Fig. 6.1: RegEx components based on their detection complexity and depth

A. Mathematical Model for Dos Filter:

In DDos filter we count number of request per 40 seconds from one IP address, if the no requests are greater than 100 then we block the IP address. Thus any human cannot send hundred requests per 40 second.
Suppose
r = request count per 40 seconds (time stamp) from particular IP.
t= time stamp which we set 40 seconds.
c = no of request any human can possibly send to server here we set (100).
If
r > c in time t
Ip address blok, (No further request from this IP processed)
If not
Then allow IP to communicate with server.

B. Architecture Diagram:

The figure 6.2 shows architecture diagram for our project. It shows client and server transaction which done through DOS filter and DPI module to avoid attack on server such as DOS attach and Sql Injection Attack.

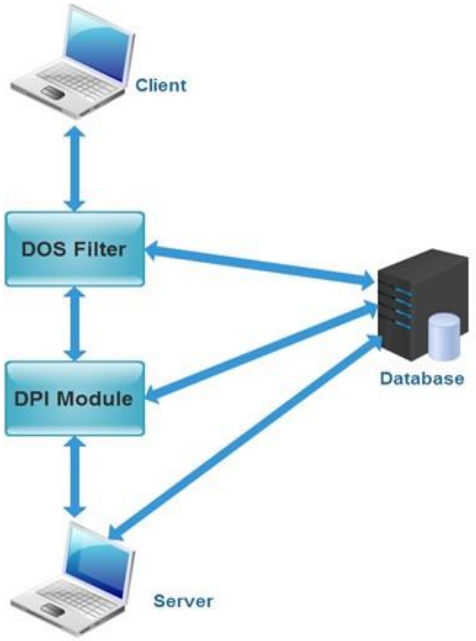


Fig. 6.2: Architecture Diagram

C. ER Diagram:

ER Diagram means Entity Relationship Diagram. The Entities are mapped to the tables in the application. An entity-relationship (ER) diagram is a UML diagram that shows relationships between entities in a database. In figure 4.2.2 client, DOS filter, DPI module, server and data base are entities. Connection line shows the relationship between them, and ovals represent attributes of entities.

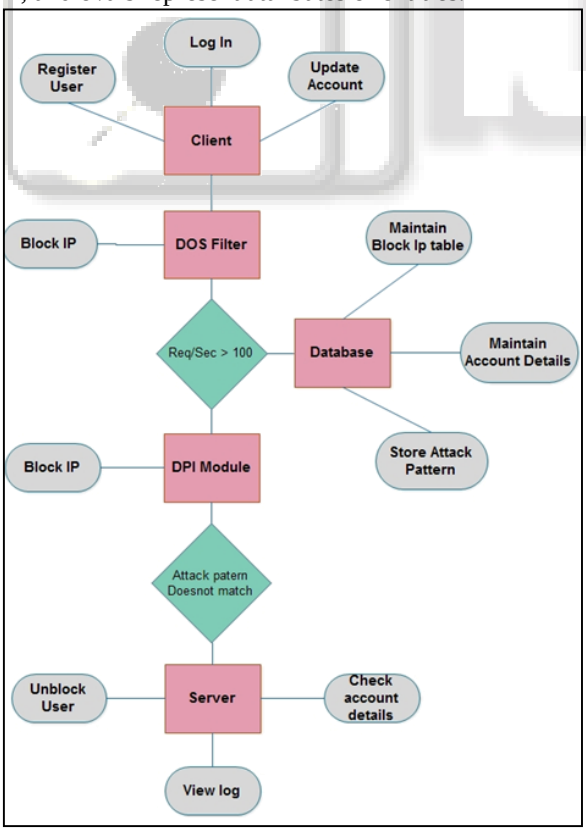


Fig. 6.3: E-R diagram for our project

D. DFD Diagram:

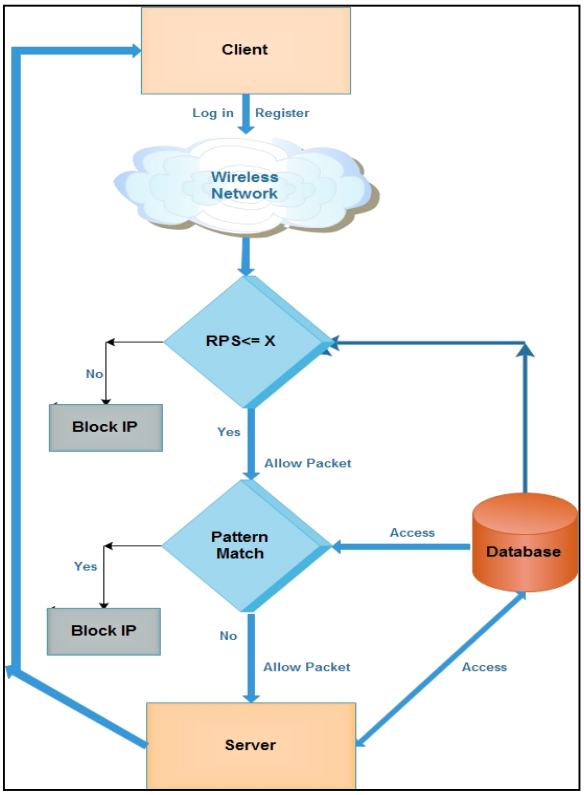


Fig. 6.4: DFD level 2 Diagram (Dpi Module)

VII. RESULTS AND COMPARISON

In this section we discuss on the result of our project. After all the test cases we test our program for various DDOS attack on our server site web page. It results in machine IP block as we expected. We test our project for Sql injection and XSS attack for testing DPI technique pattern matching by inserting various Sql injection and XSS attack, our project blocked every attack. We also tested admin privileges to unlock any client which would be block earlier. Admin can unblock any client from the database by accessing admin login.

Admin can also see the information about attack that what kind of attack occurs at which date and time, Server can store the list of attack according to its time and date

Technology	XSS attack detection time (in ms)				
	Attempt 1	Attempt 2	Attempt 3	Attempt 4	Attempt 5
LaFA	132	121	185	203	130
DPI-AD	73	72	68	77	70

Table 3: Comparison chart for XSS attack Detection between existing system and proposed System (ms).

In above comparison table:

- XSS attack attempts compare between LaFA [1] and DPI-AD
- Table values denote required time for XSS attack detection and prevention in milliseconds.
- It shows the DPI-AD required much less time for XSS attack detection compare to LaFA [1] technology.

Following figure 6.1 shows graph chart for table 4.1

- Red bar denotes XSS attack detection time (milliseconds) by LaFA [1] technology.
- Blue bar shows XSS attack detection time required by DPI-AD technology.
- It shows the DPI-AD required much less time for XSS attack detection compare to LaFA [1] technology.

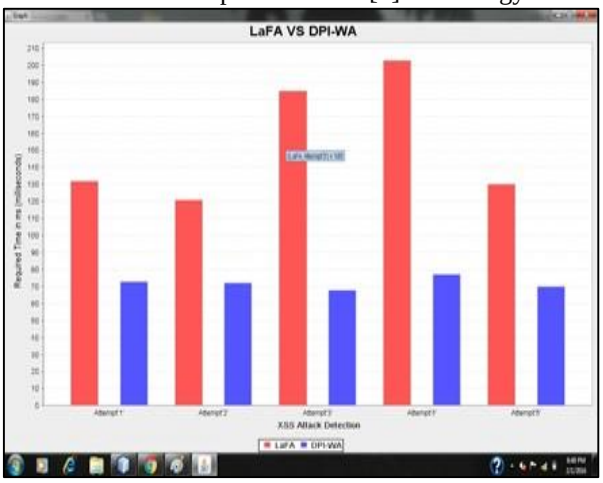


Fig. 9: Comparison Graph for XSS Attack Detection Speed between LaFA [1] and DPI-AD.

A. Advantages of DPI-AD:

DPI-AD methodology improves network intrusion detection throughput with help of DPI techniques. Dos filter remove or blocked all ddos attack malicious packet it filters the dos attack packets due to which it improve malicious packet detection.

- Speed of intrusion detection and prevention is increased.
- Authorized user can access data without any security risk.
- Memory requirement for server security is reduced.

We test our project for Sql injection and XSS attack for testing DPI technique pattern matching by inserting various Sql injection and XSS attack, our project blocked every attack. We also tested admin privileges to unlock any client which would be block earlier. Admin can unblock any client from the database by accessing admin login.

Admin can also see the information about attack that what kind of attack occurs at which date and time, Server can store the list of attack according to its time and date.

VIII. CONCLUSION AND FUTURE ENHANCEMENT

In existing systems, there are some limitations Intrusion detection and prevention system. In our project DDOS filter blocks the DDOS attacks so that the intrusion detection module only face packet without DDOS attack pattern, hence it automatically increases intrusion detection and prevention speed.

In DDOS filter we implement technique to count request per second and set the value that cannot be achieve by any human user, If the request per second count is greater than predefined value system block clients ip, so that the attacker cannot try to attack on server again. Intrusion detection module use Deep Packet Inspection pattern matching technique to detection of intrusion containing packets and prevent it.

We implement our project with the use of wireless network; we tested it with ad-hoc wireless network and internet as client server module.

A. Future Enhancement:

In future we can improve intrusion detection by using various emerging attack patterns in intrusion detection module.

REFERENCES

[1] Masanori Bando, N. Sertac Artan, and H. Jonathan Chao., "Scalable Lookahead Regular Expression Detection System for Deep Packet Inspection", IEEE Transactions on Networking, Vol. 20, No. 3, June 2012.
[2] Chad R. Meiners, Jignesh Patel, Eric Norige, Alex X. Liu, and Eric Torng., "Fast Regular Expression Matching Using Small TCAM", IEEE/Acm Transactions On Networking, Vol. 22, No. 1, February 2014.
[3] Tiwari Nitin, Solanki Rajdeep Singh and Pandya Gajaraj Singh, "Intrusion Detection and Prevention System (IDPS) Technology- Network Behavior Analysis System (NBAS)", ISCA Journal of Engineering Sciences, Vol. 1(1), 51-56, July 2012.
[4] Cong Liu, Yan Pan, Ai Chen, and Jie Wu., "A DFA with Extended Character-Set for Fast Deep Packet Inspection", IEEE Transactions On Computers, Vol. 63, No. 8, August 2014.
[5] Xiaofei Wang, Yang Xu, Junchen Jiang, Olga Ormond, Bin Liu, and Xiaojun Wang, "StriFA: Stride Finite Automata for High-Speed Regular Expression Matching in Network Intrusion Detection Systems", IEEE Systems Journal, Vol. 7, No. 3, September 2013.
[6] Anat Bremler-Barr, DavidHay, and Yaron Koral, "CompactDFA: Scalable Pattern Matching Using Longest Prefix Match Solutions", IEEE/Acm Transactions On Networking, Vol. 22, No. 2, April 2014.
[7] Tamer AbuHmed, Abedelaziz Mohaisen, and DaeHun Nyang., "A Survey on Deep Packet Inspection for Intrusion Detection Systems", Information Security Research Laboratory, Inha University, Incheon 402-751, Korea, March 2008.
[8] Klaus Mochalski, and Hendrik Schulze,"White paper on Deep Packet Inspection", ITU-T study groups com13.
[9] Fang Yu, Zhifeng Chen, Yanlei Diao, T. V. Lakshman, and Randy H. Katz, "Fast and Memory-Efficient Regular Expression Matching for Deep Packet Inspection", ACM 580-0/06/0012, December 3–5, 2006.
[10] Bing Chen, Lee, J., and Wu, A.S., "Active event correlation in Bro IDS to detect multi-stage attacks", Fourth IEEE International Workshop on Information Assurance, 13-14 April 2006.
[11] Rafeeq Ur Rehman, "Intrusion Detection Systems with Snort", ISBN 0-13-140733-3, Library of Congress Cataloging-in-Publication Data, Prentice Hall PTR Upper Saddle River, ew Jersey 07458.

