Multi-Layered Intrusion Detection Approach for Web Applications

Muhammad Sanusi[1], Sani Halima[2], Atumoshi Adamu Yusuf[3], Daniel Okunbor[4]

Senior Lecturer[1], Research Scholar[2], Research Scholar[3], Professor[4]
[1-3]Computer Science Department, University of Abuja, Nigeria
[4]Department of Mathematics and Computer Science, Fayetteville State University, Fayetteville, NC 28301, North Carolina

***Abstract:-*Web applications are becoming the dominant way to provide access to online services such as e-commas, e-banking, hospitals, school portals, etc. and also a valuable target for security attacks. As the use of web applications for some or most services has increased, the sophistication of attacks against these applications has grown as well. To protect web applications, its services and servers against web application related attacks, the multi-layered detection system is proposed in this research work. This technique proposal will enhance the level of protection considering the diverse nature of intrusions. The contemporary methods in IDS such as the Anomaly-based, Signature-based and Policy-based IDS and their challenges are also reviewed. The common web application attacks which are vulnerable to the web application where discussed. The proposed Multi-layered Technique will detect and filter malicious code irrespective of the entry point in as much the code will be encapsulated as part of the HTTP header. Furthermore, to minimize false-positive response (false alerts), the HTML contained variables are prepared before interacting with the rule. Another effort is to eliminate 'UserAgent 'from the HTTP header that would be scanned by the rules.**

Keywords: **Anomaly detection, intrusion detection, web attacks, web application firewall, web application security.**

I. INTRODUCTION

The early computer network was found when a Military research group began to investigate the large-scale coordination of digital information in the 1950's. In the late 1960's, the Advanced Research Projects Agency Network (ARPANET), attempted the integration of the first large-scale of networks and was called "internetworking", the National Science Foundation (NSF) took over and opened the Internet for commercial use in 1991 [1]. Ever since the use of computer and computer networks has rapidly grown and is an essential tool in the development of most business enterprises and organizations. Some users, do not have good intentions when accessing these systems, rendering them targets for attacks, thus, keeping them safe is important. The popularity of web applications has attracted attackers who try to exploit its vulnerabilities, 95% of the respondent organizations reported having experienced more than 10 incidents related to their web sites as in CSI/FBI Computer Crime and Security Survey of 2005 [2].

Intrusion Detection System (IDS) gives information in advance about an attack or intrusion attempts, by detecting the actions of intruders, as such becoming an important technology in the business sector as well as an active area of research [3]. The main idea behind an IDS is to detect deviations from the normal behavior of a monitored resource such as computer systems and networks. An IDS can be classified according to several features, e.g., the kind of data it analyzes (host, network or application data) and the techniques it uses to detect anomalies (signature-based, anomaly-based, expert systems, fuzzy logic, artificial intelligence, Multi-layered detection, etc.). Intrusion Detection Systems are implemented using Hardware and/or Software.

There are different IDS types and approach, they are Network-Based (NIDS), Host-Based (HIDS) and Application Based (AIDS) intrusion detection systems [5].

- *NIDS:* These collect information from the network itself rather than from each separate host. NIDS audits the network attacks while packets are moving across the network and NIDS analyze the flow of information between computers. The agents are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network.

- *HIDS:* This locates the sign of intrusion in the local system. HBIDS examines host _based actions such as what applications are being used, what file is being accessed, and what information resides in the kernel.

- *AIDS:* These checks the effective behavior and event of the protocol. The system or agent is placed between a process and group of servers that monitors and analyzes the application protocol between devices [6].

- *Signature Based:* Signature-based IDS have the capability of detecting known attacks and generate less false-positive alarms but in some cases can miss an attack if the respective pattern is not stored in the database. SIDS cannot detect a zero-day or unknown attack.

- *Anomaly Based:* The anomaly-based approach looks for behavior or use of computer resources deviating from "normal" or "known" behavior to an 'abnormal 'or 'unknown 'behavior. The underlying principle of this approach is that

"attack behavior" will differ from the "normal user behavior" thus it can be detected by cataloging and identifying the differences that occurred [7].

❖ *Policy Based:* The policy-based approach establishes boundaries between the legitimate and none legitimate activities by imposing a set of rules [8]. With this technique, a zero-day attack can be detected and the classification of normal unseen behavior into attack class is carried out.

The rest of the paper is organized as follows: Section 2 presents a literature survey where we have a study of the related works. Section 3 presents the proposed work where we have discussed the proposed work in the field of IDS. And finally, section 4 presents results and Section 5 conclusion.

II. LITERATURE REVIEW OF RELATED WORKS

The first suggested Anomaly Based IDS was in [9], where the authors described a system that uses Bayesian parameter estimation to analyze web access logs and detect anomalous sessions in web applications. In a paper presented by [10], an experiment was conducted in which the NSL-KDD data set was used to develop a new intrusion detection hybrid model with higher accuracy and performance, classifiers such as J48, Metapagging, Random Tree were used to develop the hybrid model. The experimental results revealed that the hybrid approach had a significant effect on the minimization of the computational and time complexity involved when determining the feature association impact scale. The study in [11] further carried out the context-aware anomaly detection by presenting an innovative method for representing the packet payload using contextual n-grams ($_c$n-grams). The $_c$n-grams technique allows integration of structural properties of protocol and their respective byte sequences in a unified feature space.

In the Signature Based IDS Authors in [12], [13] carried out a study by using the implementation process of Snort in Debian. Here the Snort was placed between two hosts. Her signature was first designed, which show the flow of packets, data was flown using TCP replay within two systems. The result shows that once the Snort identifies an intrusion attempt, it will send an alert to the security person and the security person takes required action immediately. Study [14], analysis log entries to recognize malicious activities on the webserver. [15] showcase WebSTAT, an intrusion detection system based on the STAT framework which can monitor both logs and HTTP requests. The system relies on the state transition analysis model to describe the attack scenarios and is also capable of detecting multi-step attacks. Moreover, it permits the detection of attacks variant which is similar to the specified malicious behavior. The IDS presented in paper [16] provides an architecture that leverages the strengths of both techniques (anomaly and signature) in such a manner that it gives the advantage of categorizing the events into safe, intrusive or unknown class (i.e., the class for which events neither qualify as an attack, nor as safe).

In the policy-based approach, establishes boundaries between normal and abnormal activities by imposing some rules [12], and also set equilibrium between allowed and not allowed events. Authors in the study [17] incorporated some set of policies such as Generic Authorization and Access control API (GAA - API) in an IDS to make the system capable of identifying an unauthorized operation. Implementation of an Extended Access Control List (EACL) language to specify security policies for monitoring resource access and acting with response to any threatening activity was carried out. A paper presented by the Authors in the study [18], indicated the ontological model as a new breed of IDS and highlighted two ontology models, namely protocol-centric and attack centric model to identify malicious requests. These provide the baseline on which the security measures of the attack model are constructed by assisting the system with inference and reasoning ability to map different scenarios of security breaches to a general semantic rule.

A multi-layered approach, here the Authors in paper [5], proposed a multi-layer intrusion detection technique (MLIDS) model. The proposed work uses an attack model to identify various types of attacks and also shows on which layer the attack was identified. An adaptive base support threshold was applied on selected axis attributes in mining the Internet episode rules

2.1 Comparison of Some Web Application IDS

IDS tools are compared based on performance metrics, such as detection coverage and false-positive rate [12]. In this research work, the comparison is based on their design methodology and the functions carried out. From the comprehensive literature review in Section 2 above, several dimensions of web application IDS have been considered to make comparisons from a different perspective such as Detection Approach, IDS Type, Data Monitored Type, and IDS Mode. Table 2.1 below shows comparison of some web application IDS

Table 2.1 Comparison Of Some Web Application IDS

Reference	Detection Approach	IDS Type	Data Monitoring Type	IDS mode	Rank
[9]	SD	HIDS	Logs	Detection	Good
[10]	SD	HIDS	Logs, Get, Post, Header	Detection	
[11]	PD	HIDS	Get, Post, Header	Prevention	Average
[12]	HD	HIDS	Get, Header	Detection	Fear
[13]	SD	HIDS	Logs	Detection	
[14]	AD	HIDS	Get	Prevention	Near Perfect
[15]	AD	NIDS	Logs, Get, Post	Detection	
[16]	PD	HIDS	Get, Post, Header	Detection	
[12]	AD	HIDS	Get, Post, Header	Prevention	
[17]	AD	NIDS	Get, Post, Header	Detection	
[18]	AD	HIDS	Get, Post, Header	Detection	
[5]	PD	HIDS	Get, Post, Header	Prevention	
MLIDS	**AD, SD & PD**	**HIDS & NIDS**	**Logs, Get, Post, Header**	**Detection & Prevention**	**Perfect**

AD: Anomaly-based Detection, SD: Signature-Based Detection

PD: Policy-Based Detection, HD: Hybrid-Based Detection

HIDS: Host-based Intrusion Detection System, NIDS: Network-based Intrusion Detection System,

MLIDS: Multilayered-Based Intrusion Detection System

Evaluation of the contributions of each attribute to their dimension was made based on the comparison of the IDS. From the detection approach dimension, works focused more on the anomaly-based IDS due to its ability to detect novel or new attacks. The policy-based IDS imposes a set of rules to establish a balance between the anomaly and signature detection approach. The IDS type dimension shows that most of the detection systems are host-based compared to network-based systems. In the data monitored type dimension, only a few of the systems are monitoring the traffic flowing out of the webserver. Monitoring the response data from the server to the client as an output of the client's request processed by the server prevents the application from exposing sensitive information unintentionally. The IDS mode dimension shows that the prevention mechanism has little less attention [19].

2.2 Web Application Attacks

The Open Web Application Security Project [20] has periodically published their findings on this subject. An attack is unauthorized access which can cause damage to computer systems, records or networks. Some of the most recent and top web application attacks are [26]:

1) Injection Strings such as SQL Injections
2) Cross-Site Scripting(XSS)
3) Insecure Direct Object References

- *SQL Injection:* This takes place when untrusted data are injected in an interpreter as part of a query, tricking it to execute unintended commands or access unauthorized data.
- *Cross-Site Scripting (XSS):* this occurs when an application takes a user-supplied data and sends it to a web browser without proper validation or encoding of the data, this enables an attacker to execute scripts in the victim's browser.
- *Insecure Direct Object References:* **A**n attacker manipulates the references to an internal implementation object such as a file. to access unauthorized data.

III. PROPOSED SYSTEM DEVELOPMENT METHODOLOGY

In this research work, an Agile development method is adopted in developing the entire system. [21]. The adopted procedure is illustrated in Figure 3 below which comprises multiple phases. The work starts with the development of a web application that consists of user interface with form elements, using HTML5, PHP and MySQL on Apache server. The second phase is the development of the proposed multi-layered IDS. The integration of the system is carried out in the third phase. The last phase is where system testing and evaluation were carried out using the data collected from online sources (Mavituna, OWASP, and Hackers, 2017, see reference).

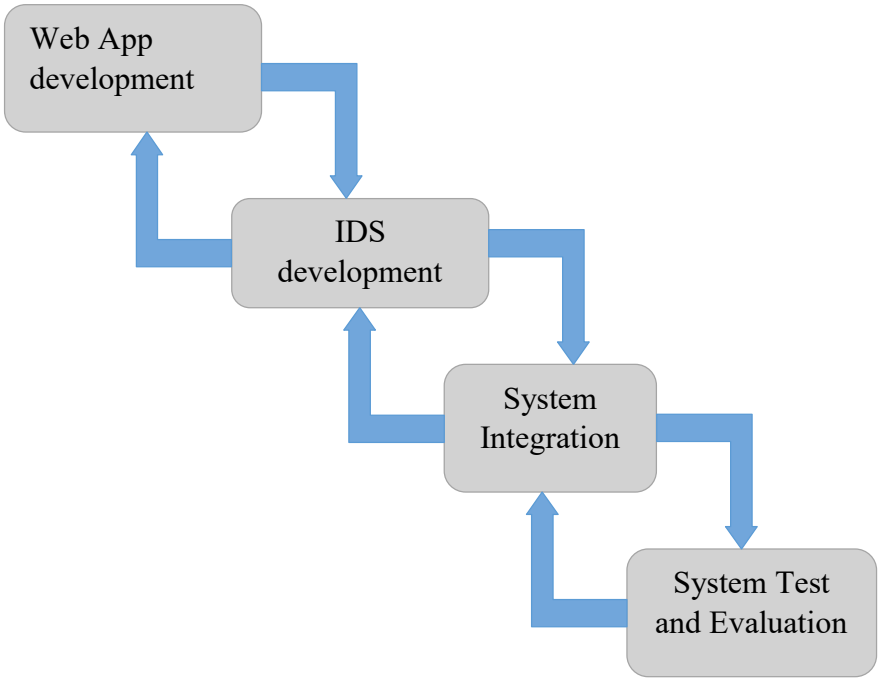

Figure 3: Research procedure

3.1 Proposed System Architecture

The proposed Multi-Layered approach consists of the Policy-Based (PD), Anomaly-Based (AD) and Signature-Based (SD) intrusion detection techniques in combination to achieve a higher detection accuracy, lower false alarms and as such, high level of Cyber trust**.** The proposed system architecture shown in Figure 3.1, the system operates as a proxy located between the client and the webserver.

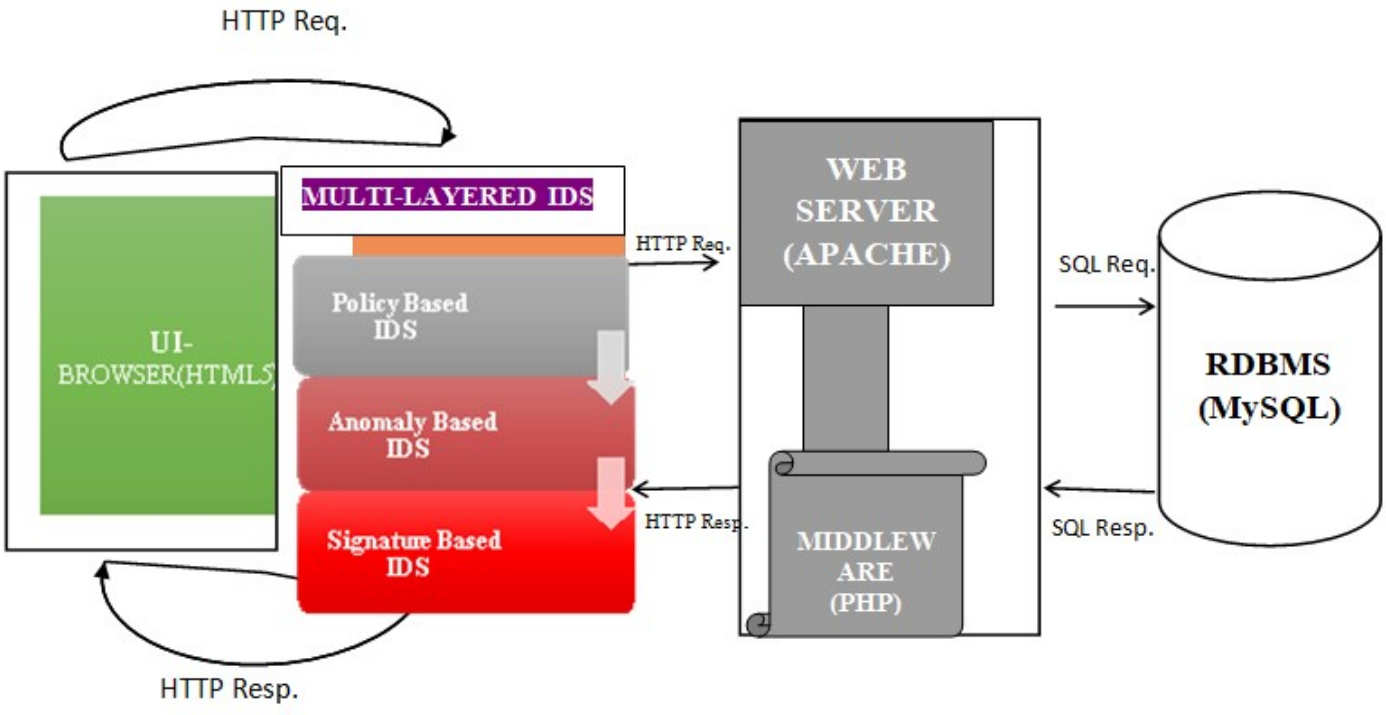

Figure 3.1: Proposed System Architecture

In the proposed multi-layered system architecture above, the first layer is the Policy-Based Approach (PB), where some set of policies where put in place such as access control were a client must have a valuable and authentic identification before accessing the web server, the second layer is the anomaly-based technique (AD), here the system operates in two different modes as a detection and prevention mode.

In detection mode, the system analyses the incoming request, trying to find suspicious patterns. If a suspicious request is detected, the proxy launches an alert; otherwise, it remains inactive. In any case, the request will reach the webserver. When operating in detection mode, attacks could succeed, whereas false positives do not limit the system functionality.

In prevention mode, the system receives requests from clients and analyzes them. If the request is valid, the system routes it to the server and sends back the received response to the client. If not, the proxy blocks the request and sends back a generic denied access page to the client. Thus, the communication between proxy and server is established only when the request is deemed valid.

The third layer which is the signature-based technique (SD), here the signatures are used to analyze previous attacks. The collected signatures stored in the database are used to match incoming client requests to detect intrusions. The signature-based technique is used to detect known attacks with low false alarms, though it cannot detect unknown or new attacks without any pre-collected signatures or lack of attack classifiers. In the proposed MLIDS it's used only for single-connection attacks.

3.2 Filtering Techniques

The proposed system needs a precise picture of what normal behavior is in a specific web application. In this research work, the proposed system relies on XML and PHP files which contain a thorough description of the web application's normal behavior. Once a request is received, the system compares it with the normal behavior model. If the difference exceeds the given thresholds, the request is flagged as an attack and an alert is launched.

To make the proposed system more effective, a double-layered technique is applied to fortify the system for detecting and filtering intrusion.

3.2.1 First-layer

The first layer is the XML file that contains rules regarding the correctness of HTTP verbs, HTTP headers, accessed resources (files), arguments, and values for the arguments.

3.3.2 Second-layer

The second layer consists of a customized module developed with PHP in-built functions for filtering. PHP filters are used to validate and filter data coming from insecure sources, like user input. By using filters, it is very sure the application gets the correct input type. All external data must get filtered before getting into the application. Input filtering is one of the most important application security issues.

3.3 WEB_IDS DATABASE: Here the database only consists of two tables as explained in tables 3.3.1 and 3.3.2 below.

3.3.1 Student table: This stores the student's registration information. The table structure is shown in Figure 3.3.1 below

#	Name	Type	Null	Extra	Key
1	id	int(11)	No	AUTO_INCREMENT	Pk
2	Matric	varchar(15)	No		
3	Surname	varchar(20)	No		
4	Other names	varchar(20)	No		
5	Phone	varchar(15)	No		
6	Email	varchar(20)	No		
7	Faculty	varchar(20)	No		
8	Department	varchar(20)	No		
9	Level	int(11)	No		
10	Gender	varchar(7)	No		

Figure 3.3.1: Student Table structure

3.3.2 Intrusions Table: The table stores the log of intrusions detected by the IDS. The structure of the table is shown in Figure 3.5.2 below

	Name	Type	Attributes	Null	Extra	Action
1	id	int(11)	UNSIGNED	No	AUTO_INCREMENT	Pk
2	Name	varchar(128)		No		
3	Value	Text		No		
4	Page	varchar(255)		No		
5	Tags	varchar(255)		No		
6	Ip	varchar(15)		No		
7	ip2	varchar(15)		No		
8	Impact	int(11)	UNSIGNED	No		
9	Origin	varchar(15)		No		
10	Created	Datetime		No		
11	domain	varchar(255)		No		

Figure 3.4.2: Intrusions Table Structure

IV. TOOLS USED FOR THE WORK ARE

A laptop was used in experimenting, all the tests conducted were performed under this computer hardware and no other background services were running. The proposed multi-layered system in this thesis was running on a Windows operating system and was developed in a virtual environment. Virtual Operating System is used to allow proper testing without affecting normal system files or other web services running on the system. Tools such as VMware Workstation 11 for virtualization, Windows 7 as client OS, Apache 2.2.12 which serves as the server, MySQL 5.1.37 for the database quarry, HTML 5 and PHP 5.4.0 for the development of the web application with form elements. All are running default configurations so that the conducted experiments can be reproduced with the same results.

4.2 Implementation Procedure

The proposed multi-layered system architecture can be viewed as an extra virtual layer that is integrated between the web server and the web application. All the traffic that flows from the client to the web application has to pass through the system. This was achieved by modifying the normal operations of a web server and the interpreter to call the proposed multi-layered IDS for each HTTP request, including form data encoded in the POST method that is received by the webserver. The system can retrieve all the field values such as Host: localhost, User-Agent: Mozilla Firefox, Cache-Control, etc. from the HTTP header and are the content that are scheduled to be scanned.

4.3 IDS Filters

In the proposed system, filters are stored in an XML file which is loaded upon program start. The filters consist of regular expressions that are used to identify harmful patterns that are used to exploit vulnerabilities. Filters are formulated as regular expressions, and a sample filter is provided in Table 4.2 below

What can be gathered from the regular expression above is that it will protect the system from the attacks explained in the description column of Table 4.2.

Table 4.2: Sample regular expressions used in the proposed IDS

o	Rule	Description
1	(?:(((.*)\%[c|d|i|e|f|g|o|s|u|x|p|n])(Amoroso)))	Looking for a format string attack
2	(?:(union(.*)select(.*)from))	Looking for basic sql injection. Common attack string for MySQL, oracle and others
3	<![CDATA[(?:"[^"]*[^-]?>)|(?:[^\w\s]\s*\/>)|(?:>")]]>	Finds html breaking injections including whitespace attacks
4	<![CDATA[(?:"+.*[<=]\s*"[^"]+") |(?:"\s*\w+\s*=)|(?:>\w=\/)|(?:#.+\) ["\s]*>)|(?:"\s*(?:src|style|on\w+)\s *=\s*")|(?:[^"]?"[,;\s]+\w*[\[\(])]]>	Finds attribute breaking injections including whitespace attacks

4.4 Test System Development and Setup

The proposed IDS is experimented using a simple web application form database enabled. The front end of the test system was developed with a combination of languages such as HTML5, PHP and MYSQL. It is a three-layer structured web application so that important parts such as the user interface, the scripting files, and the back end are separated. Additionally, a specific identifier was used to call the appropriate file based on a URL variable so that repetition can be minimized in coding.

The application allows users to register a student in which the information is stored in the database and login student. The main concern is not the type of robustness of the test application itself since the IDS can be integrated with any PHP/MySQL-powered web application, the concern is to monitor and filter the traffic between client and server system component.

4.5 Result Discussion

The proposed multi-layered system was tested with two cases as discussed below;

i) By appending malicious code directly to the web application URL in Figure 4.1 below

(http://localhost:8080/webapps/web-ids/index.php?action=../../../../../../etc/passwd)

and*(http://localhost:8080/webapps/web-ids/index.php?action=%22%3EXXX%3Cscript%3Ealert(1)%3C/script%3E)*.

The results as indicated in Figure 4.2.and 4.3 respectively shows that the codes are both malicious with (dt, id, lfi) and (XSS, csrf, id, rfe, lfi, sqli) and impart 15/32 respectively.

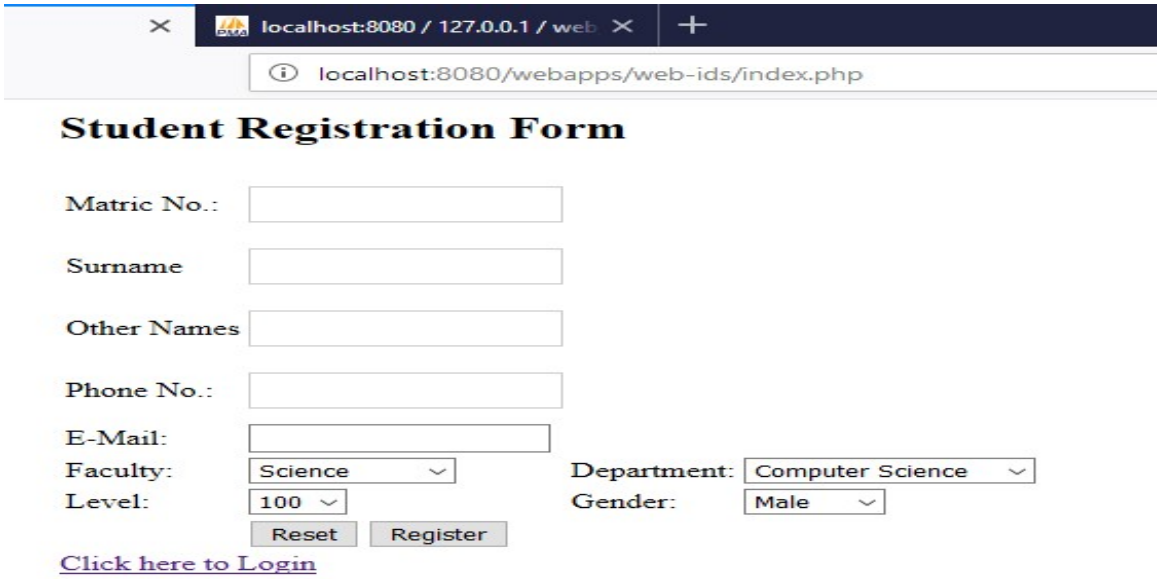

Fig 4.1: User Interphase

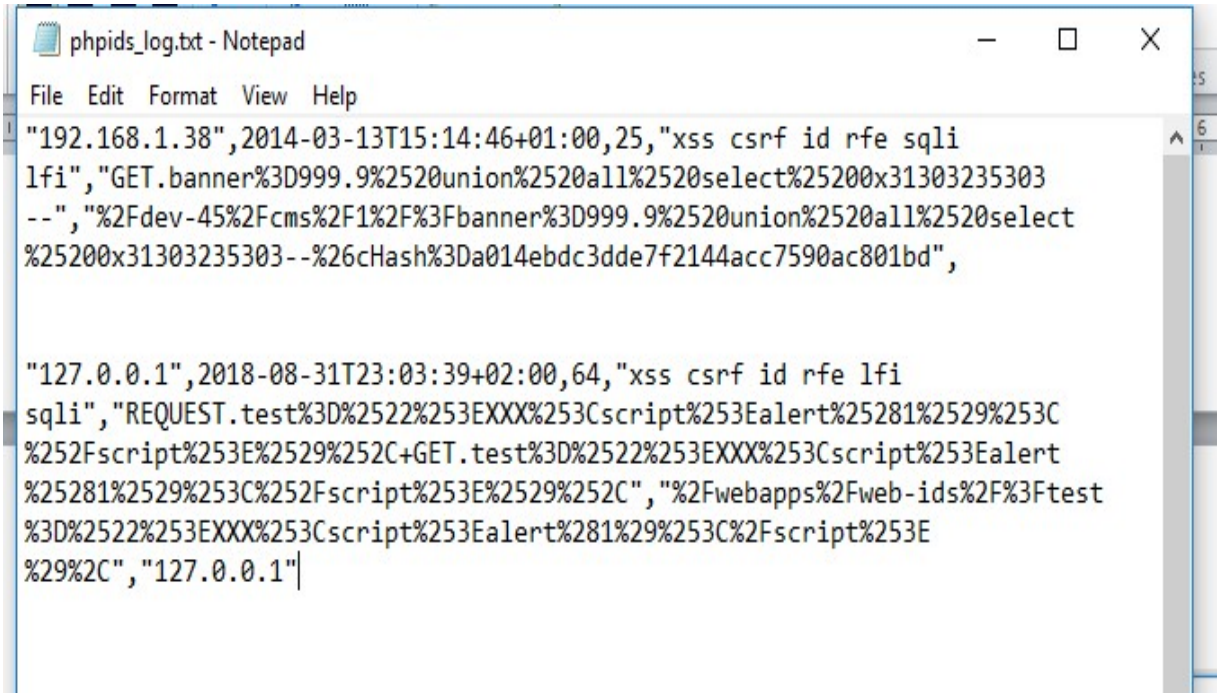

Figure 4.3: Log text file sample

ii) By typing "1/*" in the Matric textbox and "UNION SELECT*" in the Surname textbox of the same form shown in Figure 4.1 above. The result indicates that the code is malicious with name (ss, csrf, id, sqli, lfi) and has an impact on the system as shown in Figure 4.5

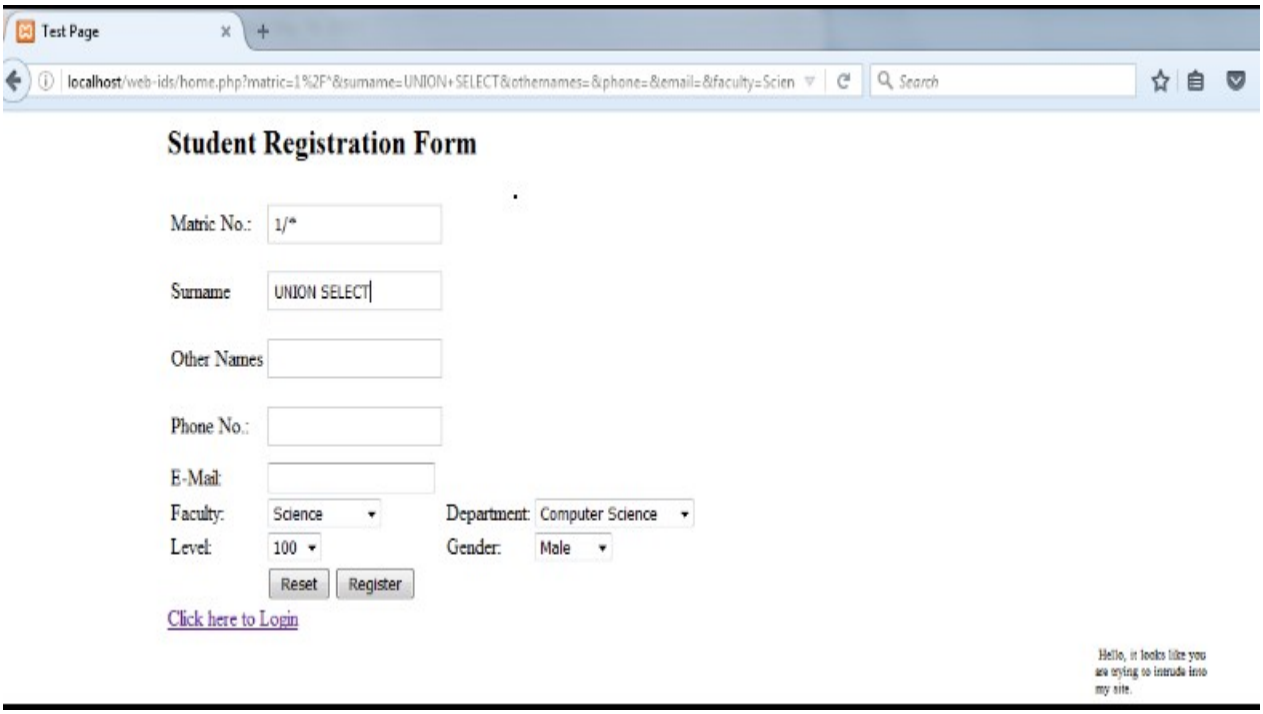

Figure 4.4: Intrusion Detection

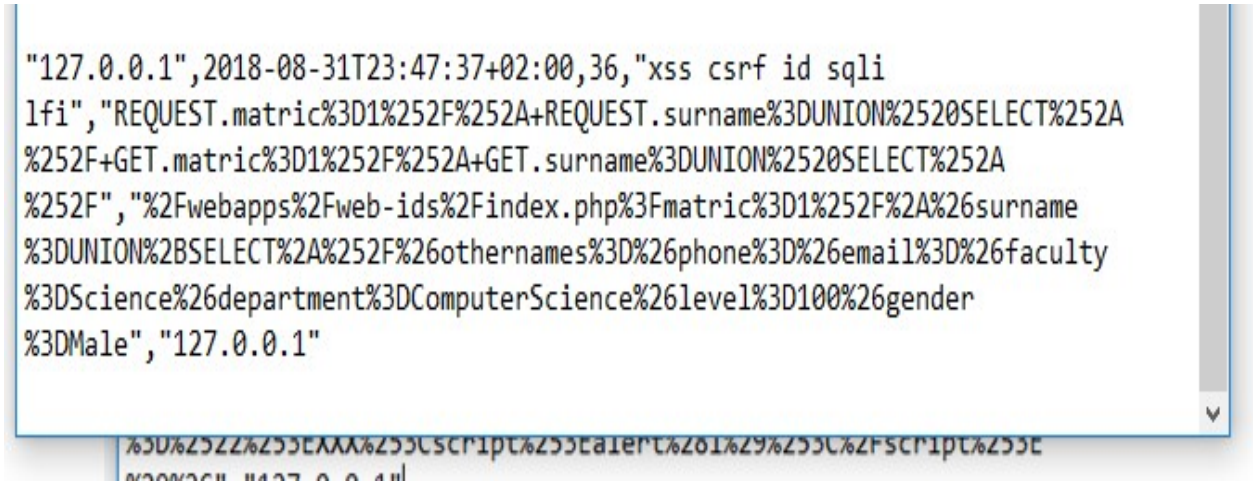

Fig 4.5: Log Table

V. CONCLUSION

In this research work, a Multi-layered IDS is introduced. This technique detects and filters malicious code irrespective of the entry point, several methods are reviewed. The technique further disallows false-positive response alerts as the multi-layering includes various variables are considered. The technique rules are employed to eliminate user agents from the HTTP header. Future work wills implementation of the proposed Multi-layered approach in full-fledged web application model so this can detect attacks in real-time intrusion detection environments.

REFERENCES

[1]. Akinwumi, A., Olutayo, A., Samuel, O., & Olujimi, A. (2017). Laying Foundation for SCADA System Protocol Performance Modelling. *Journal of Network Communications and Emerging Technologies (JNCET) www. jncet. org, 7*(11).

[2]. Torrano-Giménez, C., Perez-Villegas, A., & Alvarez Maranón, G. (2010). An anomaly-based approach for intrusion detection in web traffic.
[3]. Gordon, L. A., Loeb, M. P., Lucyshyn, W., & Richardson, R. (2005). 2005 CSI/FBI computer crime and security survey. *Computer Security Journal, 21*(3), 1.
[4]. Kumar D., Venugopalan S. (2017) Intrusion detection systems: a review. Int J Adv Res Comp Sci. 2017; 8(8):356–370.
[5]. Agrawal, G., Kamble M. (2012) Proposed Multi-Layers Intrusion Detection System (MLIDS) Model. International Journal of Computer Science and Information Technologies (IJCSIT) Vol.3, pp.5040–5042.
[6]. Vijayarani, D. S., &Jothi, M. P. (2014). Hierarchical and Partitioning Clustering Algorithms for Detecting Outliers in Data Streams. *International Journal of Advanced Research in Computer and Communication Engineering, 3(4)*, 6205-6207.
[7]. Torrano-Giménez, C., Perez-Villegas, A., & Alvarez Maranón, G. (2010). An anomaly-based approach for intrusion detection in web traffic.
[8]. Pałka, D., & Zachara, M. (2011, August). Learning web application firewall-benefits and caveats. International Conference on Availability, Reliability, and Security (pp. 295-308). Springer Berlin Heidelberg.
[9]. Cho, S., & Cha, S. (2004). SAD: web session anomaly detection based on parameter estimation. *Computers & Security, 23*(4), 312-319.
[10]. Al-Sarawi, S., Anbar, M. Alieyan, K., Alzubaidi, M. (2017) Internet of Things (IoT) communication protocols: Review. In Proceedings of the 2017 8th International Conference on Information Technology (ICIT), Amman, Jordan, 17–18 May 2017
[11]. Duessel, P., Gehl, C., Flegel, U. et al. (2016). Detecting zero-day attacks using context-aware anomaly detection at the application-layer. International Journal of Information Security, 1-16.
[12]. Nancy, A., & Syed, Z. H. (2018), A closer look at Intrusion Detection System for web Applications.
[13]. Vinod, K., Dr. Prakash, O., (2012) Signature-based intrusion detection system using Snort" International Journal of Computer Applications & Information Technology Vol. I, Issue III, (ISSN: 2278-7720)
[14]. Magnus A., Ulf L., and Erland J., (2008) A multi-sensor model to improve automated attack detection. In *11th International Symposium on Recent Advances in Intrusion Detection (RAID 2008)*. RAID, September 2008.
[15]. Vigna, G., Robertson, W., Kher, V., et al. (2003, December). A stateful intrusion detection system for world-wide-webworld-wide-web servers. In Computer Security Applications Conference, 2003. Proceedings. 19th Annual (pp. 34-43). IEEE.
[16]. Stuttard, D., & Pinto, M. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. John
[17]. Ryutov, T., Neuman, C., Dongho, K. et al. (2003). Integrated access control and intrusion detection for web servers. IEEE transactions on parallel and distributed systems, 14(9), 841-850.
[18]. Razzaq, A., Latif, K., Ahmad, H. F. et al. (2014). Semantic security against web application attacks. Information Sciences, 254, 19-38.
[19]. Stassopoulou, A., & Dikaiakos, M. D. (2009). Web robot detection: A probabilistic reasoning approach. Computer Networks, 53(3), 265-278.
[20]. OWASP, T. (2017). Top 10-2017 The Ten Most Critical Web Application Security Risks. *URL: owasp. org/images/7/72/OWASP_Top_10-2017_% 28en, 29*.
[21]. Sommerville, I. (2011). Software engineering 9th Edition. *ISBN-10137035152*.
[22]. Amoroso, E. G. (1999). *Intrusion detection*: Intrusion. NetBooks.
[23]. McHugh, J. (2001). Intrusion and intrusion detection. International Journal of Information Security, 1(1), 14-35.
[24]. Razzaq, A., Ahmed, H. F., Hur, A. et al. (2009, February). Ontology-based application-level intrusion detection system by using a bayesian filter. In Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on (pp. 1-6). IEEE.
[25]. Murphy, M. (2016). No place to hide as DNS comes under attack. Network Security, 2016(7), 5-7.
[26]. OWASP, T. (2017). Top 10-2017 The Ten Most Critical Web Application Security Risks. *URL: owasp. org/images/7/72/OWASP_Top_10-2017_% 28en, 29*.