Modern Applied Science; Vol. 12, No. 10; 2018
ISSN 1913-1844E-ISSN 1913-1852
Published by Canadian Center of Science and Education

Distributed Detection and prevention of Web Threats in Heterogeneous Environment

Saher Manaseer[1], Ahmad K. Al Hwaitat[1] & Riad Jabri[1]

[1] Department of Computer Science, King Abdullah the II IT School, The University of Jordan, Amman, Jordan

Correspondence: Department of Computer Science, King Abdullah the II IT School, The University of Jordan, Amman, Jordan. E-mail: saher@ju.edu.jo

Received: December 12, 2017 Accepted: December 30, 2017 Online Published: September 9, 2018

doi:10.5539/mas.v12n10p13 URL: https://doi.org/10.5539/mas.v12n10p13

Abstract

The growth of web Applications have increased rapidly due to the huge development of technology with very short turnaround time and with this development the protection from vulnerabilities became very difficult. There is a continuous demand for developing new methods that is able to prevent the fast growth of attacking methods and vulnerabilities. Furthermore there is a great demand to have coordination between different security infrastructure and protection applications to distribution of the attack log in order to prevent the attacker from further attacks to other web hosts. This research proposes a distributed web firewall defensive mechanism which provide a synchronized environment that is consists of several synchronized web application firewalls. Every web application is protected by a web application firewall that send feedback reports that include the type of the attack, The IP Address of the attacker and time of attack to other synchronized firewalls inside the environment to take action against the attacker.

Keywords: SQL Injection, XSS, DDoS Attack, Suspicious User Behavior, Web Applications, Distributed System, Firewalls, IDS

1. Introduction

Recently, the web became the main link that connects all users all over the world where data about the web users are stored in databases (Rajan et al., 2010). Some of these activities contain sensitive and private information about the user such as credit card numbers, passwords, and money authorization transactions information. The security of the user's information is a major concern for all company owners and administrators because of the existence of successful attacks against web applications across the history. Many attackersmay be able to compromise some web applications and access private data across the global net by exploiting several known and un-known web application vulnerabilities (Al-hamami etal., 2012). Therefore, there is a major need for developing researches and find methods for preventing and detecting any possible attack against such web based infrastructure, securing databases and help making the data more private for the users. Furthermore, there is a need to take pre-steps and create a method to detect and prevent such harmful activities before even taking place through the Internet by distributing the attacker's information to other firewalls as a new approach to web application security field.

The current method of defending and preventing attacksuses standalone web application firewalls (WAFs).In order to change this whole approach the need of building a distributed firewall systemthat sharesthe attacker'sinformation to other web applications firewalls in-order to take a defensive procedures against the same attacker and the same type of attack before the time of occurrence on other web based targets, which will provide an extra layer of security against known types of attacks for more than one target at the same time.

This paper aims to find a solution for the challenges and limitations in confidential information protection. In addition, to improve the prevention and detection methods of web attacks through the proposed heterogeneous distributed firewalls. We also intend to achieve better interactivity and performance to protect web applications from malicious users, and prevent them from injecting malicious web content, which took advantage of the vulnerabilities that developers overlooked. To achieve this goal, we used the open-source web application firewall. Firewalls take a set of steps to prevent attacks by banning the IP address of an attacker among the shared hosts, then broadcasting the attack information such as type of attack, attacker IP Address, and time of attack with all of shared hosts.

2. Related Works

A new technique was proposed by (Dr R.P Mahapatra, 2012) to keep java web applications secure from cross site scripting attacks (XSS) and the is by making a framework on pattern matching approach. The components of the framework are request/response analyzer and a modifier models, the foundation of this study was that framework could be implemented on a web application which already exists, and on manually crafted web applications. The strong aspect of their framework was the ability of implementing it on any existing jave web application, and no source code modification is needed.

A system which verified all kinds of SQL injection attacks, was developed by (Pratik et al, 2014), the result of their method is preventing stores procedure attack and cross site scripting attack (XSS). All input strings, which are responsible for the query implementation and analyzing them fully, have been recorded.

Al-SharifS.et. al, (2016) They presentedaround testing on computers by people belonging from different walks of life like graduate and undergraduate students, students and faculty. The initial test's result demonstrates that it was be utilized as an effective administration and also as a device to assist with endorsing awareness against a variation of users that are focused mainly on cyber-attacks especially phishing attacks.

(O. Oriola, 2006) considered agent and data mining independently and mutual benefits. A theoretical concept and framework based on integrating multi-agent system peer-to-peer computing for distributed data mining the system implemented in Java Agent Development Environment. The theoretical concept has been proposed for the DIDS is peer-to-peer agent mining integration

A new algorithmic attempt was made by (Garg S., Narula P, 2014), presented to detect SQL injection attacks and avoid them early. The technique was performed by using unclear parameters and some rules. The technique they use consist of the steps mentioned next, collection data set which is meant for training, and also a cheat sheet for the analysis, then data set must be trained for the interrogation, generation number 3 of the patterns and keys, fourth is that each parameter gets a full analysis, the implementation of the proposed model on the training set of data, the final step is fetching the results and data interpretation. The results they got from the work they did were not bad regarding security, and the time they spend on executing the script.

Karam Y. et. al (2012) They proposed a model named Secured Objective-Driven programming which is created automatically at runtime by PAA Cloud Engine with also the XACML security annotation representation. The latter provides for a secured, separated abstraction layer made exclusively for the cloud users who sat on top of the programming model.

A research conducted by Balasundaram and. Ramaraj in 2011 presented a novel specification-based technique to prevent attacks of SQL injection. Aims to first, preventing all forms of attacks of SQL injection; and second, the fact that the method used currently doesn't give the user access to the database directly in database server.

Manasser S. and Hwaitat A.(2017) They a proposed of an enhanced web application solution has been made, which is a valid and integral element that can be simply installed on any WAF-Web Application Firewall. This app tries to solve the problem mentioned and elevate the standard of trust between web application's hosts/owners and users. Further, it will help in restoring the loss of data that is caused by hacking efforts in simple and organized manner.

A mechanism was developed by (Hidhaya1 2012), for the detection of SQL injection. By employing a Reverse proxy and MD5 algorithm to watch SQL injection in input. Using rules of grammar expressions for checking SQL injection in URL's. No changes are done in the application's source code by their method. Investigating and decreasing the attack is automatically done. The increasing in the number of proxy servers makes web applications able to handle any number of requests with no delay of time, and makes it able to protect the application from SQL injection attack.

A mechanism which has both static and dynamic analysis was developed by (Balasundaram, Ramaraj, 2011) in the stage of static analyzing, the method of prevention was first displayed in a level of three phases: viperous text detector, field constrain validation, and last, structure Query Languagevalidation. In the stage on runtime validation, the data which was the user's input is checked (validated) with all the mentioned stages, and the result shows if the input is safe or not

A novel and an effective solution for the problem of XSS were suggested by (Bangre and Jaiswal, 2012), its purpose was to detect all SQLIA kinds. Their technique also checked the assigned value for a single quote, double dash and space given by the user through input sections. A space, single quotes or double dashes are ought to be used by the attacker in his input, when he is scripting an SQL injection.

A prevention technique against DDoS attack on REST based web service, was presented by (Lad and Baria, 2014), in this technique, resources were represented by a special URL that was generated by a part of the core set of HTTP orders: Put, Post, Get and Delete. Web services which are REST based perform DDoS easily. It'll monitor the behavior of the IP address, by employing a number of requested URL and time Interval Analysis which is based on threshold.

A system of binary protection intrusion was proposed by (Venkatesh, 2014), the purpose of which is designing the behavior of the network of the user sessions, and that is across the front-end web server and back-end database. The rummaged out attacks, that wasn't distinguished by separate IDS, by checking both web and database requests.

3. Method

3.1 Distributed Web Firewall System

The Enhanced web firewall system mechanism flowchart is shown in figure 1 and the general idea of the distributed firewall and the way that the firewalls are connected one to the other is shown in figure 2.

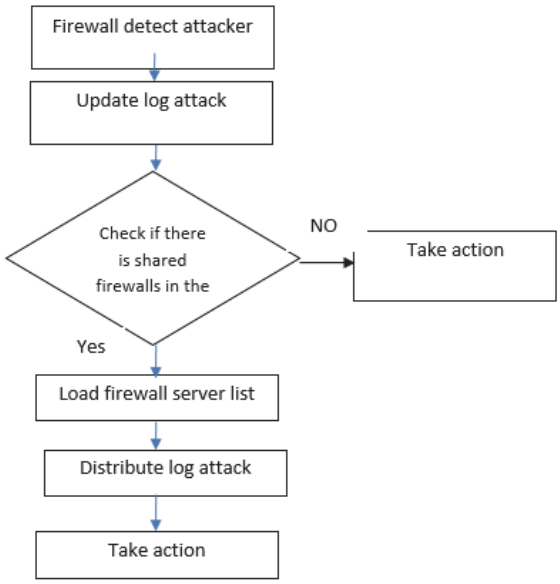

Figure 1. Enhanced web firewall system mechanism flowchart

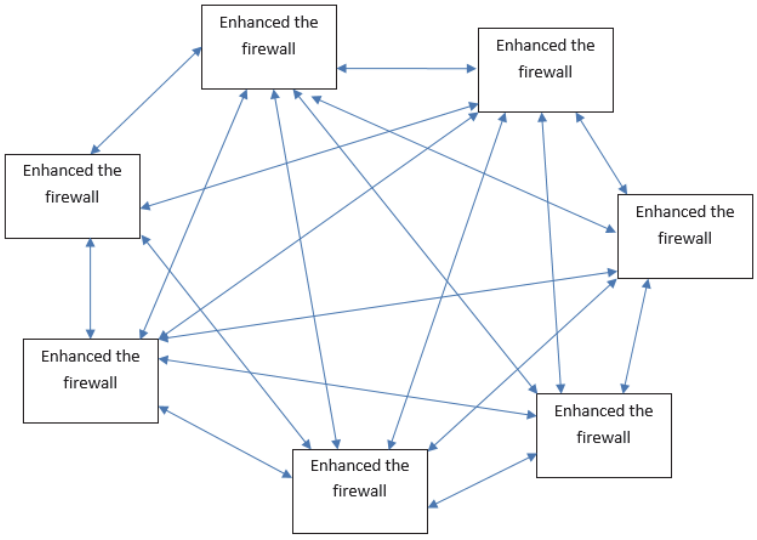

Figure 2. Distributed Web Firewall systems

3.2 Web Firewall Core Component

The core component of any WAF consists of known procedures that have to be done before data reach the server backend. The firewall as any known type of WAF will intercept the traffic between the user and the web application in order to check whether the data is malicious or safe to be passed down to the web application and continue the interaction between the user and the web application safely and heavily monitored (Sakthivel et al.,2012). The first step in this component is analyzing the user request which is done by capturing the received data from the user then analyzing the submitted data through HTTP-HTTPS protocols by com-pairing the captured data with known signatures stored in a signature database in-order to decide whether to take actions against the user based on this operation of analysis of the captured data or let the user surf safely while being monitored (Wanget al., 2005). The reason of analyzing the HTTP-HTTPS submitted data is that the HTTP-HTTPS analysis can give a clear indication of the user behavior as a normal or a malicious act. The second component contains the logging mechanism, which logs all the events that occurs under the firewall observation.

When an attack or a malicious input is detected The administratorwill receive an Email Notification which will include the attack info mentioned before and the permission to Take counter measuressuch as (Displaying Warning message – Blocking the attacker IP- the range of IP's in the estimated region – flush the session that the attacker is using) immediately after thealert processis done, without the user's responsethose actions will be taken automatically and the web page output will be changed for the attacker as a sensitization mechanism to Compress any error caused by the attacker's actions. The sensitization mechanism is used instead of rejecting or blocking the malicious input to the web-app. and it changes the malicious input/output into an acceptable format.

The distributed Web Firewall is an enhanced Web Firewall function that has added a new value to the services of the firewall. This functionality makes the enhanced firewall different from the typical firewalls, designed to be a real time feedback sharing process, the connected firewalls works as a one circle with no danger of a one point failure. the component has been implemented with graphical user interface compatible with the Firewall, this component includesthe features to add URL, check URL, load log attack, distribute the log attack, receive the log attack, store log attack, update the local log attack in the received Web Firewall.

When the Web Firewall detect an attack through analyzing the user request which is done by analyzing the received data from the user based on analyzing the HTTP-HTTPS Interceptor and HTTP head analysis, it stores the attack information in the log file and identifies many types of attacks such as SQL injection, XSS, DDoS.Then create a newlog file whichcontains the following information; type of attack, date and time andIPAddressof the attack/attacker will be shared with the other distributed firewalls on thedistribution system see the following figure 3.

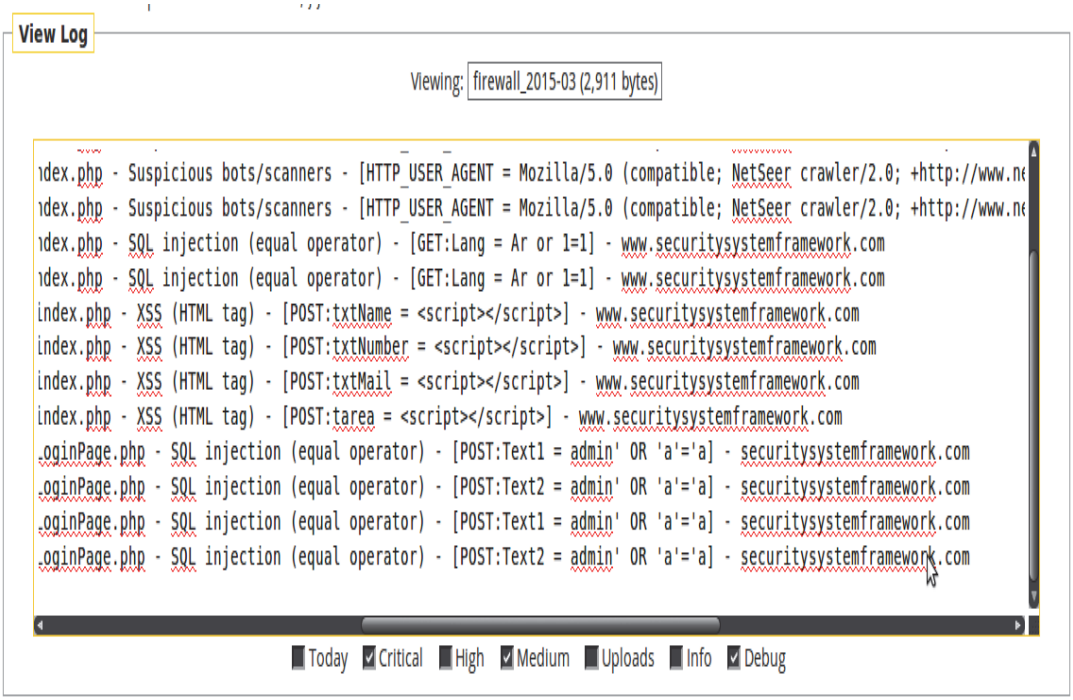

Figure 3. Log Attack Process

3.3 The Process of Adding Firewalls Server List

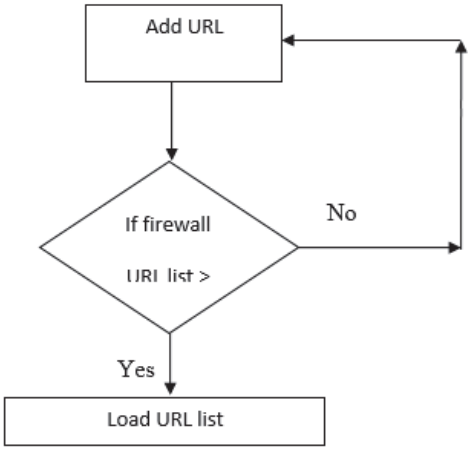

Figure 4. Load Firewalls Server List

The distribution process consists of checking the existence of the distributed servers list. The URL list should be always updated for each Web Firewall that use the Web Firewall this list should be managed by the web applications administrators.

If the Web Firewall detect an attack it automatically load the URL listof the other serversand starts a loop process in-order to send the attack/attacker's information to the synchronized firewalls see the following figure 4.

4. Experiment One SQL Injection "Command or Query"

The attacker's hostile data can trick the interpreter into injecting commands that are unintentional or retrieving data without valid authorization. For example, an SQL injection that modifies the "id" parameter value by the attacker in the browser to send: (or "1" = 1) in the SQL query will change the meaning of both queries to return all the records from the accounts table. Amore dangerous attacks, it could modify data or even invoke stored procedures.

To test SQL injection query on a sample website, we perform the query as illustrated in Figure 5. HTTP request that is sent to server. The firewall will take that request and read the payload of the received packet in order to check it using the input checker component. In this experiment the firewall check found that the URL contains (“1” = 1) characters and matches the entry in the database signature. And this mean that the HTTP request contains an illegal signature which indicates that there is a threat for the website and here the firewall will store the illegal attack, it will store the number of attack attempts, the IP Address, date and time and country of the attacker in the log file.

We use a tool named burp suite to view the submitted data as shown in Figure 6.Sending the malicious signatures in the HTTP request will be stored in the firewall log file. Repeating the SQL injection attack twice the user will trigger the alert component and response component will be broadcasting a message to all sharing firewalls of possible attack in order to take the suitable action for this attack. The firewalls will take an action to block the IP address of the attacker and ban it from accessing the website. In order to control the attack and keep the website working through preventing the attacker from attacking the database that can be accessed through SQL injection.

Figure 5. SQL injection query

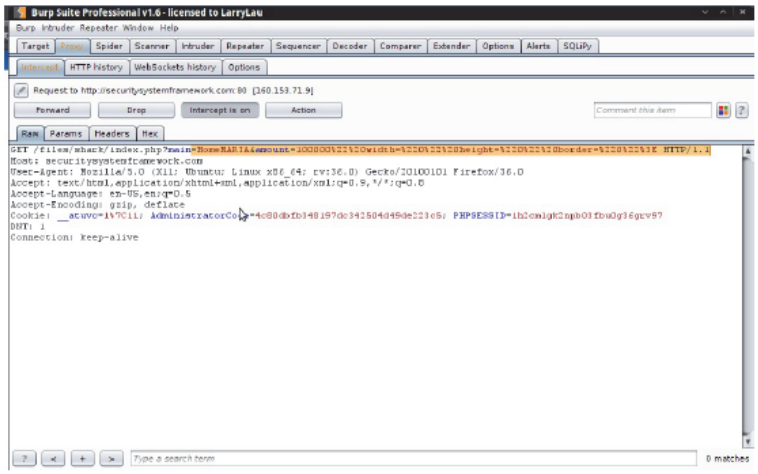

Figure 6. Burp suite HTTP analysis

4.1 Testing the Distributed Heterogeneous Firewalls

The aim of testing is to experiment the ability of the distributed heterogeneous firewall to respond to different types of attacks. Further it aims to test its ability to send the log attack to other heterogeneous Firewalls and also the ability of the other web Firewalls to receive the sent log attack and to take action based on the received log attack. Moreover the testing aims to test the Distributed heterogeneous firewalls performance.

We have used the SQL map open-source tool to test the SQL injection as shown in figure 7, an SQL injection attack has been tested to the domain(securitysystemframework.com), the web Firewall to detect the attack in the log and it will take action such as banning the IP Address as shown in figure 8. All the log actions that have been done are stored in the log attack as shown in figure 9.

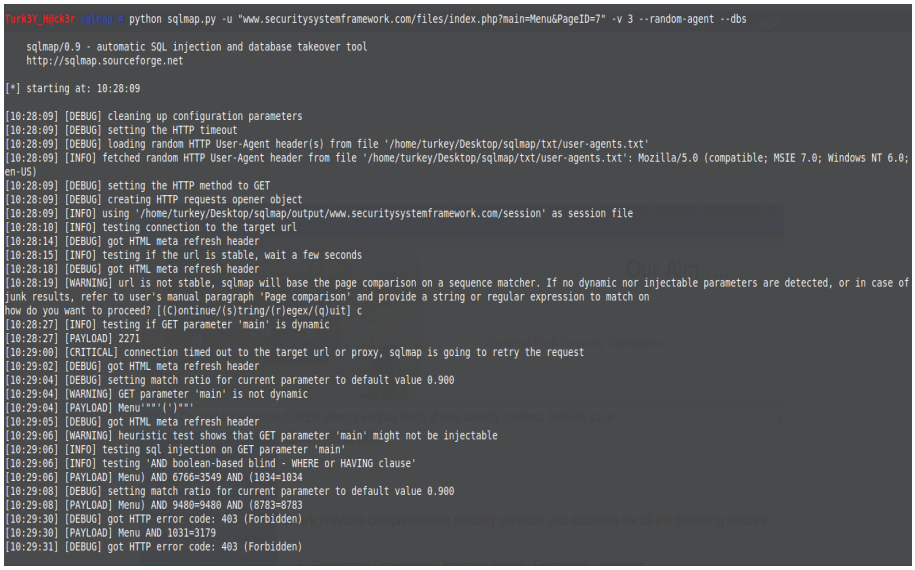

Figure 7. The SQL map tool to test the SQL injection on the sample website

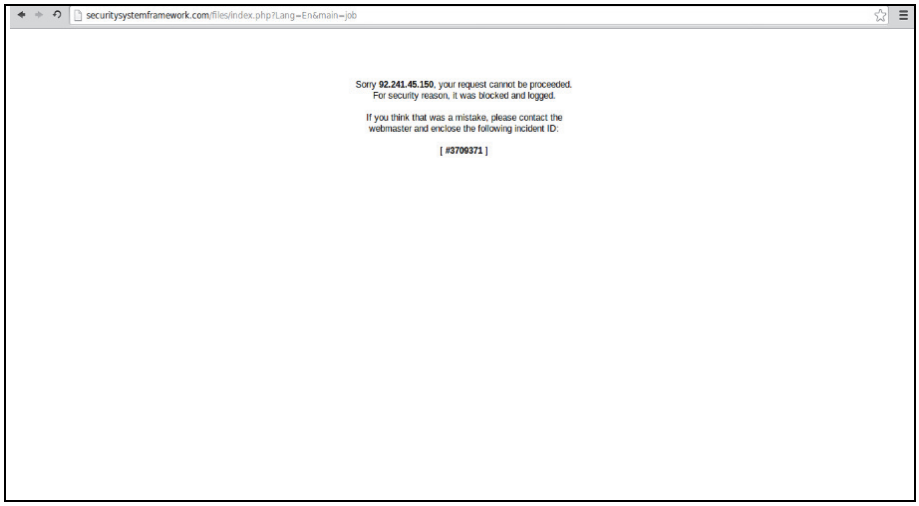

Figure 8. Banning IP Address

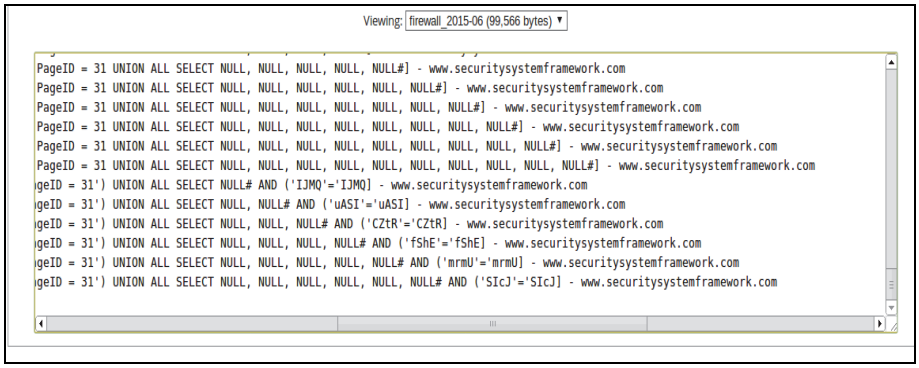

Figure 9. Log actions that has been done are stored in the log attack of the sender web Firewall

All the information that is stored in the log attack then will be sent to the second the web Firewallfound on the URL list which is (http//:www.distributedFirewallonline.com) see figure 10.

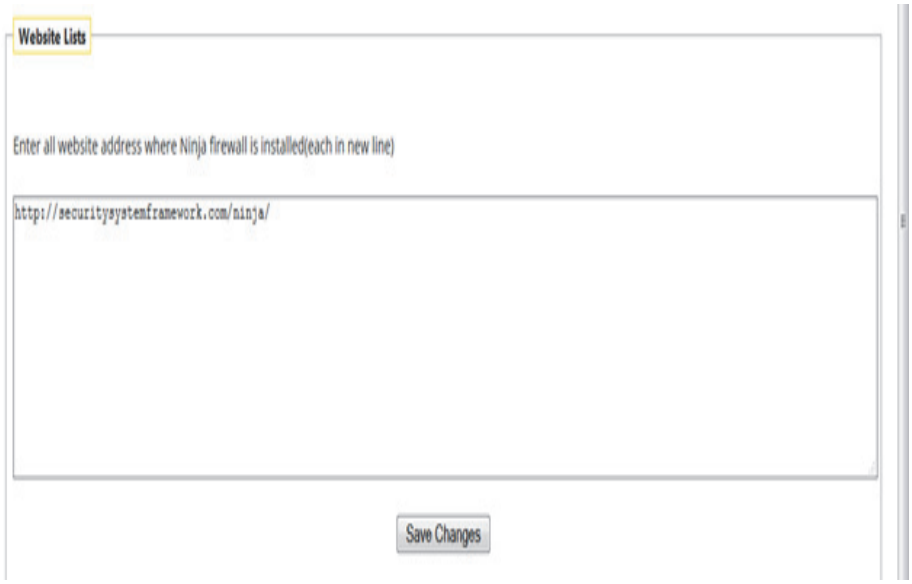

Figure 10. URL list of installed the web Firewall

A screen shot for both of the domain's log attack are shown in figure 11, figure 10 respectively.

Examining figure 12which is the log attack of the (distributedwebFirewallonline.com).We can see that it contain information about the attack that has happen to the (securitysystemframework.com) domain, it contain the type of attack, the time and date.

Since the web Firewall take actions based on the log attack then the web Firewall on the (distributedwebFirewallonline.com) will also ban the IP Address of the attacker on the (securitysystemframework.com) based on the information sent from it. And by doing this step it will be protected from possible attack by the same attackers whom have attacked (securitysystemframework.com).

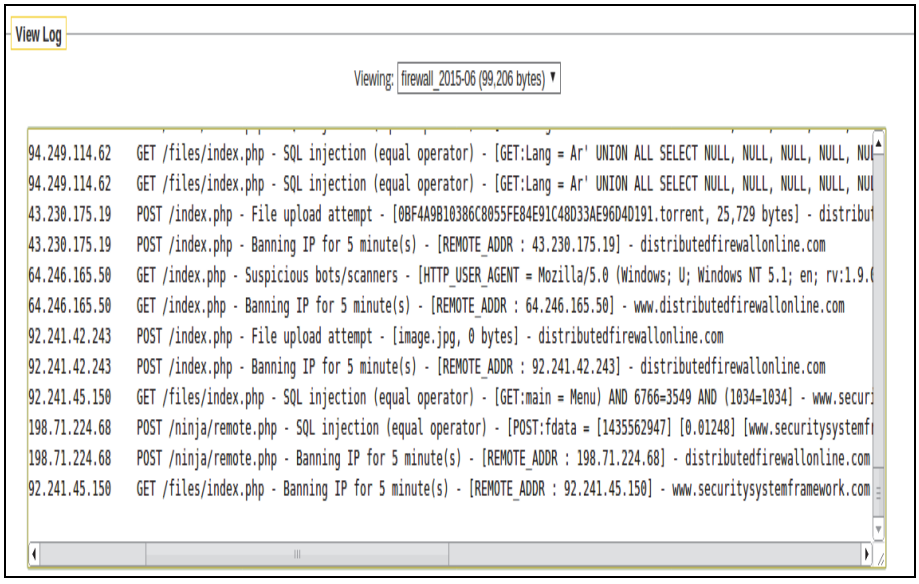

Figure 11.Web Firewall log attack in for the testing domain (http//:www.securitysystemframework.com)

Figure 12.Web Firewall log attack in for the testing domain (http//:www.distributedwebFirewallonline.com)

The log file on the received web Firewall shows the domain that has been attacked, The time of the attack, The sending time of the attack, The receiving time of the attack and the name of the domains that has received it, The type of the attack, The IP Address of the attacker.

5. Conclusions

This research presented a component which will help to enhance the performance of web application firewalls and reduce the possibility of the attack events occurrence for a varying number of hosts at the same time. The enhanced component and phases are explained to show the process of interaction between the administrator and the installed component. This research provided prevention and detection mechanism through the distribution process f web threats in heterogeneous environment. Further it provided a Distributed web firewall for detection and prevention of different types of attack. The proposed firewall is applied on Real-time distribution during the time of attack. Testing the web firewall has been experimented using web firewall with different types of attacks SQL injection, DDOS and XSS. The results showed that the proposed distributed web firewall is efficient in preventing the attack while sharing the attacker's information on other firewalls automatically in-order to take actions before the attacks occurs on them.

References

Al-hamami, A., Najadat F., & Abdul, W. M. (March 2012). Web Application Security of Money Transfer Systems. *Journal of Emerging Trends in Computing and Information Sciences, 3*(3), 365 -372.

Al-Sharif S., Iqbal F., Baker T., & Khattack A. (2016). White-Hat Hacking Framework for Promoting Security Awareness. 2016 8th IFIP International Conference on New Technologies, Mobility and Security.nov 2016. https://doi.org/10.1109/ntms.2016.7792489

Balasundaram, I., & Ramaraj, E. (2011). An Approach to Detect and Prevent SQL Injection Attacks in Database Using Web Service. *International Journal of Computer Science and Network Security, 11*(1), 197 -205.

Bangre, S., & Jaiswal, A. et al. (June 2012). SQL Injection Detection and Prevention Using Input Filter Technique. *International Journal of Recent Technology and Engineering (IJRTE), 1*(2), 145-150.

Garg S., & Narula P. (2014). A novel Approach And Implemention Of Secured Algorithm Against Sql Injections. *International Journal of Enterprise Computing and Business Systems, 4*(1), 1-7.

Hidhaya, S., & Angelina, G. (2012). Intrusion Protection against SQL Injection Attacks Using a Reverse Proxy. *Recent Trends in Computer Networks and Distributed Systems Security Communications in Computer and Information Science, 335,* 252-263.

Karam Y., Baker T., & Taleb-Bendiab. A. (2012). Security Support for Intention Driven Elastic Cloud Computing. Computer Modeling and Simulation (EMS), 2012 Sixth UKSim/AMSS European Symposium on Date of Conference: 14-16 Nov. 2012, Valetta, Malta. https://doi.org/10.1109/EMS.2012.17

Mahapatra, R. et al. (June 2012). A Pattern Based Approach to Secure Web Applications from XSS Attacks. *International Journal of Computer Technology and Electronics Engineering (IJCTEE), 2*(3), 196-203.

Manseer, S., & Hwaitat, A. (2017). Validation and Integrity Mechanism for Web Application Security. *International Journal of Engineering Research & Science, 3*(11), 34-38.

Neha, L., & Baria, J. (2014). Ddos Prevention on Rest Based Web Services. *International Journal of Computer Science and Information Technologies, 5,* 7314-7317.

Oriola, O. (March, 2012). Distributed Intrusion Detection System Using P2P Agent Mining Scheme. *African Journal of Computing & ICT, 5*(2), 3-10. https://doi.org/10.5121/ijnsa.2013.5305

Pratik, S., & Gheewala, J. (2014). Detection and Prevention of SQL Injection Attacks. *International Journal of Engineering Development and Research, 2*(2), 2660-2666. https://doi.org/10.4236/cn.2016.83017

Rababah, O., AL Hwaitat, K. A., & Manaser, S. (2016). Web Threats Detection and Prevention Framework. *Communications and Network, 8,* 170-178. https://doi.org/10.4236/cn.2016.83017

Rajan, S., Sinha, A., & Singh, J. (May 2012). Efficient Utilization of DBMS Potential in Spatial Data Mining Applications –Neighborhood Relation Modeling Approach. *International Journal of Information and Communication Technology Research, 2*(5), 465-470.

Sakthivel, A., Sankarasubramanian, R., & Velusamy, R. (June 2012). Web Application testing with eValid. *International Journal of Computer Applications, 48*(21).

Venkatesh, C. et al. (2014). Binary Protector: Intrusion Detection in Multitier Web Applications. *International Journal of Engineering Trends and Technology, 10*(2), 145-150.

Wang, X., Abraham, A., & Smitha, S. (2005). Intelligent web traffic mining and analysis. *Journal of Network and Computer Applications, 28*(2), 147–165. https://doi.org/10.1016/j.jnca.2004.01.006