ISSN (ONLINE):2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Special Issue.25, February 2017

An Anomaly-Based Approach for Intrusion Detection in Web Traffic

Jyothsana.L.P[1] Anushya.E[2], Mrs Sheela Shantha Kumari[3]
[1, 2] student, [3]Assistant Professor
[1, 2] Department of Computer Science and Engineering,
Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Avadi, Chennai-62.
[1]jyothspadma@gmail.com, [2]anushyaelumalai@gmail.com

***Abstract*:** Intrusion detection system is a system which is used for web attacks. It enables a protocol of anomaly-based approach, thus predictable and unpredictable attacks can be detected. A request which is given by a client will be verified and if an abnormal behaviour is the result then it is known as attack. The main objective of the system is to protect a real web application. In order to train the system training requests have been used. An anomaly-based network intrusion detection techniques are an efficient technology which plays a major role in the protection of target systems and networks against malicious activities. However, despite the variety of detection methods described in the literature in the recent years, security detection machines or toolswhich incorporate anomaly detection function are just introducing, and thus many
Important problems remain to be solved.

Key terms: IDS, IPS, WAF, CRLF, XSS, MODSECURITY, SQL, IIS,

1. *Introduction*

There are many IDS systems that are available primarily based on the signature detectors. Despite effective detecting known intrusion attempts and exploits, they fail to compensate new attacks and carefully processed variants of old attacks. These attacks can be regularly detected by detecting the network packet headers, or monitoring the network traffic connection attempts and session behaviour. The way to detect and defend the malicious activity are protected with the help of certain procedure such as “signatures” or “thumbprints” that are developed by human experts or by semi-automated means from known prior bad worms or viruses. It is very difficult to solve “zero-day” worm problem however; the first new unleashed worm or exploit.A zero-day (also known as zero-hour or 0-day or day zero) vulnerability which is an undisclosed computer software vulnerabilities that hackers can exploit and also leads to exploitation of computer programs, data, additional computers or a network without the knowledge of the user. It is known as a "zero-day" because it does not report publicly or announce before becoming active ,and thus it results to leaving the software's author with zero days in which it create exploitation , patches or adverse effect to the system and workarounds to implement its actions.

2 *Literature survey:*

This concept is based on a system called anomaly based Intrusion detection system which is used for detecting and preventing web attacks using certain systems called IDS,IPS,CRLF,WAF,IIS.

3 *Modules:*

3.1 Web applications

Web applications are usually categorized into three logical tiers: presentation, application and storage. Generally, a web server is the first tier that is (presentation), and machine using some web information

technology is the middle tier that is (application logic) and finally, the concept related to the database is the third tier (storage). Some of the major examples are: IIS and Apache which are popular web servers, Web Logic Server and Apache Tomcat are the applications servers which are well known and finally, Oracle and MySQL are databases which are used frequently. The web server sends requests to the application tier, which services them by making queries and updates against the database and generates a user interface.

3.2 Web Attacks

Web attacks can be categorized as static or dynamic, that depends on how they are common to all web applications which are hosted on the same platform or dependent on the particular application. Static web attacks deals with security vulnerabilities in the web application platform: web server, application server, database server, firewall, operating system, and third-party components, such as, cryptographic modules, payment gateways, etc. These security pitfalls is comprised of well-known vulnerabilities and erroneous configurations. There are both commercial and free automated tools which are capable of scanning a server in search of such known vulnerabilities and configuration errors. One of the common feature of all vulnerabilities is that they request pages, file extensions, or elements which do not form a part of the web application as intended for the end user. Thus it is very easy to detect suspicious behaviour after anyresource which does not belong to the application requested by the user .Dynamic web attacks request only legal pages of the application but they overthrow the expected parameters.The input arguments can be manipulated and leads to several attacks with different consequences such as disclosure of information, information about other users theft, command execution, etc. In this case, theinformation
Provided with the regarding the type and range of values are expected as a user input, it is possible to detect any request trying that manipulates the normal inputs.

3.3 Web Vulnerabilities

Large amount of web vulnerabilities are presented in this section. The attacks listed are included in by wasp open web application security project, which presents the most critical web application security vulnerabilities.

Obsolete file existence. Obsolete files reveals the information about the application and shows the facilities to access to the server data.

- *HTTP method validity.* In this modification of the application takes place in such a way that, these methods should never be available for an attacker.
- *CRLF injection.* It is the control characters used by operating systems to indicate the end of a line, it is possible to execute illegal commands in the system.
- *Failure to restrict URL Access.* An application only protects sensitive functionality by preventing such a way that the display of links or URLs to unauthorized users. Thus attackers can use this weakness in order to access and perform unauthorized operations by accessing those URLs directly.
- *Cross site scripting (XSS).* It is done by validating and encoding the supplied data XSS allows attackers to execute java script in the victim's browser which hijacks the user sessions and also web sites which possibly introduce worms, etc.
- *SQL injection. :* The application is occurred in the database layer of an application. The vulnerability is present when user input is either incorrect or filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed in the database.
- *Broken Authentication and Session Management.*
  Account credentials and session tokens are not properly protected. Attackers compensate passwords, keys, or authentication tokens to analyse other users' identities.
- *Remote administration flaws.*The web applications implements administrators in order to access the site using a web interface. If these administrative functions are not carefully protected then an attacker can gain full access to all aspects of a site.
- *Information Leakage and Improper Error Handling.*

ISSN (ONLINE):2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Special Issue.25, February 2017

Applications unintentionally leak information about their configuration, internal workings, or violate privacy through a variety of application problems. Then Attackers can use this weakness in order to steal sensitive data, and also conduct more serious attacks.*WAF* is the tool which is used to detect the worms. Christo Ananth et al. [3] discussed about creating Obstacles to Screened networks. In today's technological world, millions of individuals are subject to privacy threats. Companies are hired not only to watch what you visit online, but to infiltrate the information and send advertising based on your browsing history. People set up accounts for Facebook, enter bank and credit card information to various websites. Those concerned about Internet privacy often cite a number of privacy risks events that can compromise privacy which may be encountered through Internet use. These methods of compromise can range from the gathering of statistics on users, to more malicious acts such as the spreading of spyware and various forms of bugs (software errors) exploitation.

3.4 Architecture

The WAF is used to analyse HTTP requests sent by a client browser which tries to get access to a web server. The analysis takes place at the application layer. The system follows the procedure of the anomaly-based approach, detecting predictable and unpredictable web attacks, in contrast with existing signature-based WAFs .One of the popular signature-based WAF.

In the architecture, the system operates as a proxy between the client and the web server. The system may also be embedded as a module within the server. The first approach leads to the advantage of being independent of the web platform.

The proxy analyses all the traffic sent by the client. The input used for the detection process is a collection of HTTP requests {$r1$, $r2$, run}. The outputwhich is a single bit aij or each input request RI that indicates if the request is normal or anomalous. The proxy is able to work in two different modes of operation such as an IDS or an IPS.

In detection mode, the proxy analyzes the incoming packets which tries to find suspicious patterns. If a suspicious request is detected, the proxy enables an alert; or it remains inactive

In prevention mode, the proxy gets the requests from clients and analyse them. If it has valid request, the proxy routes to the server, and returns the received response to the client else 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 if when the request is valid.

A diagram of WAF's architecture is shown in Fig. 1.

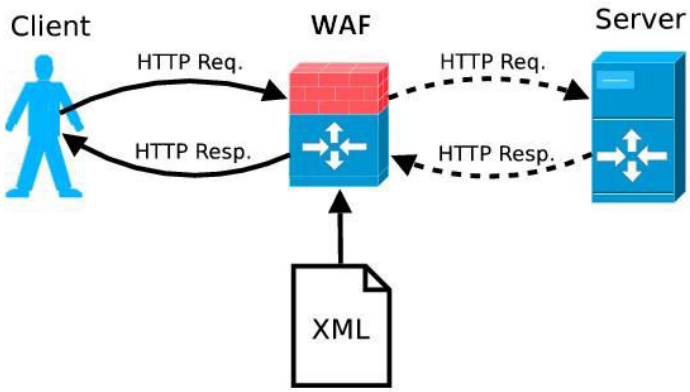

Figure 1. Web Application Firewall architecture

3.5 Detection Process

In the detection process, our system follows an approach of the form "deny everything unless it is explicitly allowed", which is also known as *positive* security model.

ISSN (ONLINE):2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Special Issue.25, February 2017

Themajor detection process takes place in the proxy. It consists of several steps, each constituting a different line of defends, in which the different parts of the request are checked with the aid of the XML file. If an incoming request fails to pass one of these lines of defends, an attack is assumed: an error page is returned to the user and the request is logged for further inspection. It is important to stress that these requests will never routes to the web server when operating in prevention mode.

The detection process is composed of the following steps:

1. Verb check. The verb must be present in the XML file, else the request is rejected.
2. Headers check. If the header appears in the XML file, its value of headers must be included too. Different values will cannot be accepted, thus it prevents the attacks
3. Resource test. The system checks if the requested resource is valid. For this purpose, the XML configuration file contains complete list of all files which are allowed to be served. If the requested resource is not present, a web attack is present.
4. Arguments test. If the request has an argument, the following aspects are checked:

a) It is checked that all the arguments are allowed for the resource. If the request includes arguments that are not listed in the XML file for the appropriate resource then manipulation of the arguments is assumed and thus the request is rejected.

Confirmation of all mandatory arguments are required. If any mandatory argument (required Field="true") is not present in the request, it is rejected.

b) Argument values are checked. An incoming request will be allowed if all parameter values are identified as normal. Argument values are decoded before being checked. For the argument value test, the statistical properties of the corresponding argument are used. If any property of the argument is outside the corresponding interval or contains any forbidden special character, the request is rejected.

These steps allow the detection of both static attacks, which request resources that do not belong to the application, and dynamic attacks, which manipulate the arguments of the request. Figure 3 depicts the detection process.

In the architecture, the system operates as a proxy between the client and the web server. The system may also be embedded as a module within the server. The first approach leads to the advantage of being independent of the web platform.

The proxy analyses all the traffic sent by the client. The input used for the detection process is a collection of HTTP requests $\{r_1, r_2, r_{un}\}$. The outputwhich is a single bit a_ij or each input request R_I that indicates if the request is normal or anomalous. The proxy is able to work in two different modes of operation such as an IDS or an IPS.

In detection mode, the proxy analyzes the incoming packets which tries to find suspicious patterns. If a suspicious request is detected, the proxy enables an alert; or it remains inactive

a) In prevention mode, the proxy gets the requests from clients and analyse them. If it has valid request, the proxy routes to the server, and returns the received response to the client else 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 if when the request is valid. It is checked that all the arguments are allowed for the resource. If the request includes arguments that are not listed in the XML file for the appropriate resource then manipulation of the arguments is assumed and thus the request is rejected.

Confirmation of all mandatory arguments are required. If any mandatory argument (required Field="true") is not present in the request, it is rejected.

b) Argument values are checked. An incoming request will be allowed if all parameter values are identified as normal. Argument values are decoded before being checked. For the argument value test, the statistical properties of the corresponding argument are used. If any property of the argument is outside the corresponding interval or contains any forbidden special character, the request is rejected.

These steps allow the detection of both static attacks, which request resources that do not belong to the application, and dynamic attacks, which manipulate the arguments of the request. Figure 3 depicts the detection process.

ISSN (ONLINE):2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Special Issue.25, February 2017

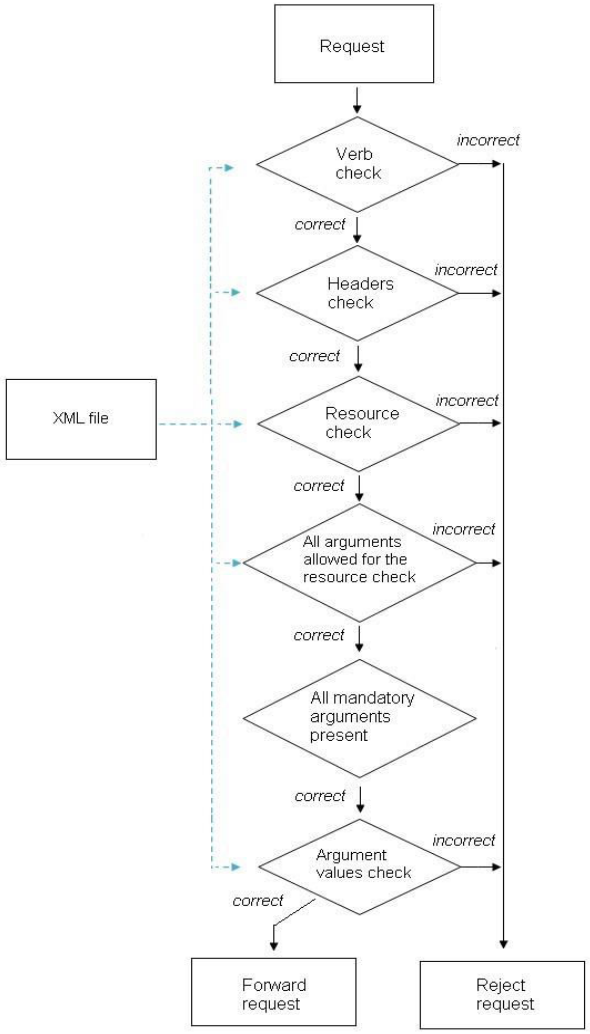

Figure 3. Detection process flow

4 Experiments

4.1 Case Study: Web Shopping

The WAF has been configured to protect a specific web application, consisting of an e-commerce web store, where users can register and buy products using a shopping cart.

4.2Artificial Traffic Generation

In our approach, normal and anomalous request databases are generated artificially with the help of dictionaries.

Dictionaries. Dictionaries are data files which contain real data to fill the different arguments used in the target application. Names, surnames, addresses, etc., are examples of dictionaries used.

A set of dictionaries containing only allowed values is used to generate the normal request database. A different set of dictionaries is used to generate the anomalous request database. The latter dictionaries contain both known attacks and illegal values with no malicious intention.

Normal Traffic Generation. Allowed HTTP requests are generated for each page in the web application. Arguments and cookies in the page, if any, are also filled in with values from the normal dictionaries. The

result is a normal request database (*Normal dB*). Some requests from *Normal dB* will be used in the training phase and some others will be used in the test phase.

Anomalous Traffic Generation. Illegal HTTP requests are generated with the help of anomalous dictionaries. Examples of the attacks trying to exploit the vulnerabilities listed in Sec. 2.3 are included in the anomalous traffic in order to test the system. There are three types of anomalous requests:

1. **Static attacks** fabricate the resource requested. These requests include attacks like obsolete files, configuration files, default files, etc.
2. **Dynamic attacks** modify valid request arguments: SQL injection, cross-site scripting, invalid parameters, command injection, buffer overflows, broken authentication and session tampering, etc.
3. **Unintentional illegal requests**. These requests should also be rejected even though they do not have malicious intention.

The result is an anomalous request database (*Anomalous dB*), which will be used only in the test phase.

4.2 Training Phase

During the training phase, the system learns the web application normal behaviour.

- Argument values are characterized by extracting their statistical properties from the requests.
- Verbs, resources and certain headers found in the requests are included directly in the XML file as allowed elements.

4.3 Test Phase

During the test phase, depicted in Fig. 4, the proxy accepts requests from both databases, *Normal dB* and *Anomalous dB*, and relies on the XML file to decide whether the requests are normal or anomalous. Considering the amount of correctly and incorrectly classified requests, the performance of the system can be measured and the results obtained.

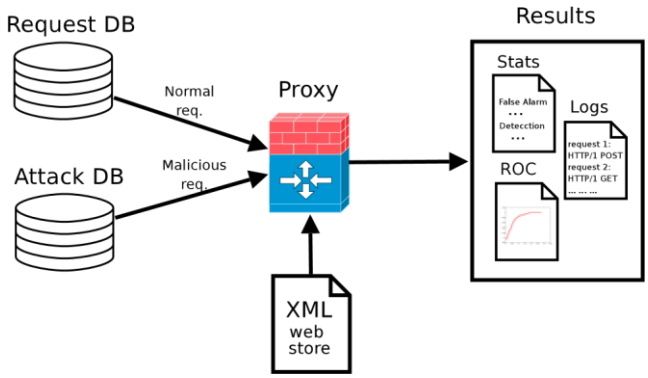

Figure 4. System test phase

4.4 WAF Protection Mechanisms

As previously stated, attacks trying to exploit the vulnerabilities described in Sec.2.3 have been included into the malicious traffic (Sec. 4.3) and they have been used to test the performance of the system.

When the normal behaviour is correctly identified, all the previously mentioned attacks can be detected (and therefore all the vulnerabilities protected) by the WAF presented in this paper. This section shows how the characteristics and mechanisms used by the WAF are effective to protect against these attacks. The WAF's mechanisms can detect both static and dynamic attacks.

When allowed directories and files are completely specified, the system protects against third party misconfiguration and known vulnerabilities. Attacks against these vulnerabilities are usually very well documented and publicized. They rely on requesting resources present by default in web servers which a

ISSN (ONLINE):2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Special Issue.25, February 2017

legitimate user will never request directly and thus are easy to spot. Therefore, the directories and files enumeration can prevent from attacks exploiting obsolete file existence, default file or example file existence, server source file disclosure, HTTP method validity, failure to restrict URL access, web application and server misconfiguration, etc.

Attacks manipulating parameters are fenced off by the proper definition of the statistical intervals. In the case of buffer overflow, the length property is of paramount importance. Many attacks make use of special characters (typically different from letters and from digits) in order to perform malicious actions. For instance, this is the case of SQL injection, which uses characters with special meaning in SQL to get queries or commands unexpectedly executed.

4.5 Performance measurement

The performance of the detector is then measured by Receiver Operating Characteristic (ROC) curves [20]. A ROC curve plots the attack detection rate (true positives, *TP*) against the false alarm rate (false positives, *FP*).

$$TveDetectionRate = \frac{}{Tve + FNve} \quad (1)$$

$$FP\ FalseAlarmRate = \frac{}{Fve + TNve} \quad (2)$$

The parameter of the ROC curve is the number of requests used in the training phase.

4.6 Results

Several experiments have been performed using an increased amount of training requests in the training phase. For each experiment, the proxy received 1000 normal requests and 1000 attacks during the test phase.

Figure 5 (a) and (b) show the results obtained by the WAF while protecting the tested web application. As can be seen in Fig. 5 (a), very satisfactory results are obtained: the false alarm rate is close to 0 whereas the detection rate is close to 1. As shown in Fig. 5 (b), at the beginning, with a low amount of training requests, the proxy rejects almost all requests (both normal and attacks). As a consequence, the detection rate is perfect (1) and the false positive rate is high. As the training progresses, the false alarm rate decreases quickly and the detection rate remains reasonably high.

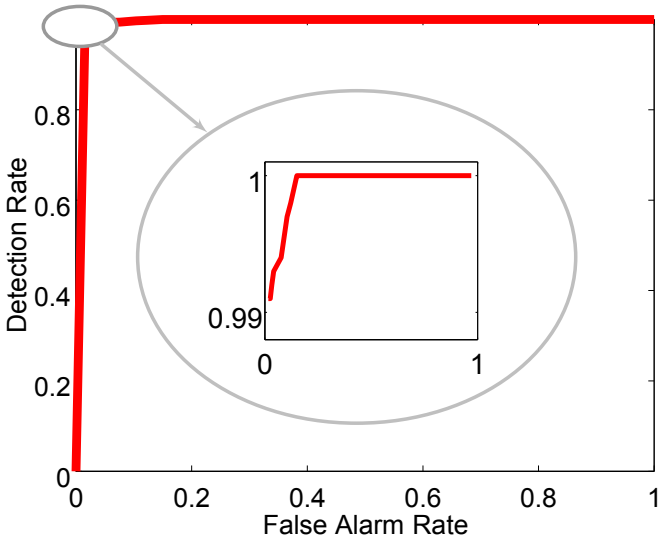

Figure 5 (a). ROC curve of WAF protecting the web store

ISSN (ONLINE):2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Special Issue.25, February 2017

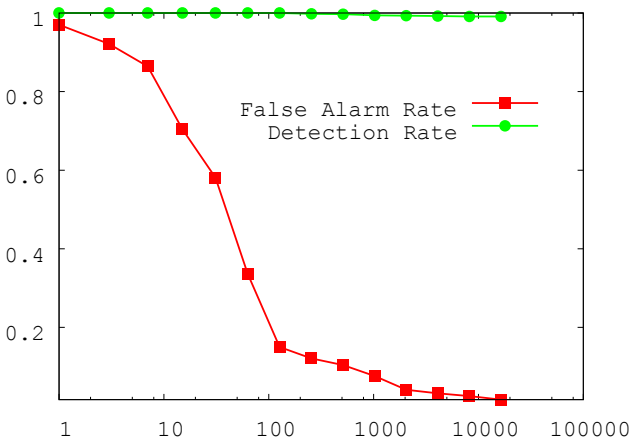

Figure 5 (b). The false alarm rate and the detection rate vs the number of training requests is plotted

5 Conclusions

We presented a simple and efficient web attack detection system or Web Application Firewall (WAF). As the system is based on the anomaly-based methodology it proved to be able to protect web applications from both known and unknown attacks. The system analyzes input requests and decides whether they are anomalous or not. For the decision, the WAF relies on an XML file which specifies web application normal behavior.

The experiments show that as long as the XML file correctly defines normality for a given target application, near perfect results are obtained. Thus, the main challenge is how to create an accurate XML file in a fully automated manner for any web application. We show that inasmuch great amounts of normal (non-malicious) traffic are available for the target application, this automatic configuration is possible using a statistical characterization of the input traffic. Future work refers to signing cookies and hidden fields in order to avoid cookie poisoning and hidden field manipulation attacks. Also, URL patterns will be used in describing sites with dynamic resources.

REFERENCES

[1] García-Teodoro P., Díaz-Verdejo J., Maciá-Fernández G., Vázquez E.Anomaly-based network intrusion detection: Techniques, systems and challenges, Computers and Security, 28, 1-2, 18-28 (2009)

[2] Patcha, A., and Park J.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks. 51, 12, 3448–3470 (2007)

[3] Christo Ananth, P.Muppidathi, S.Muthuselvi, P.Mathumitha, M.Mohaideen Fathima, M.Muthulakshmi, "Creating Obstacles to Screened networks", International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Volume 1,Issue 4,July 2015, pp:10-14

[4] Pouget F., Dacier M., Zimmerman J., Clark A., Mohay G.: Internet Attack Knowledge Discovery via Clusters and Cliques of Attack Traces. Journal of Information Assurance and Security, 1, 21-32 (2006)

[5] Estévez-Tapiador J., García-Teodoro P., and Díaz-Verdejo J.: Measuring normality in HTTP traffic for anomaly-based intrusion detection. Computer Networks, 45, 2, 175–193 (2004)