Attack Detection and Prevention Techniques in Web-based Applications

Gopal Chandak
Department of Computer Engineering
K.J. Somaiya College of Engineering
Mumbai, India
gopal.c@somaiya.edu

Tanay Dusane
Department of Computer Engineering
K.J. Somaiya College of Engineering
Mumbai, India
tanay.dusane@somaiya.edu

Zohair Irani
Department of Computer Engineering
K. J. Somaiya College of Engineering
Mumbai, India
zohair.irani@somaiya.edu

Gajanan Bherde
Department of Computer Engineering
K.J. Somaiya College of Engineering
Mumbai, India
gajananbherde@somaiya.edu

Abstract—*Web applications are becoming a part of our day to day life. People are using web applications for buying vegetables to financial transactions in their everyday lives. Lots of users can carry out their transactions, get information, and communicate by accessing these applications. Due to increase in number of users, the web applications are required to engage in organizations' or users' sensitive data. This sensitive data if not protected is worth millions of dollars to adversaries. Security in web applications is often an overlooked aspect. This is the reason why most of the organizations are susceptible to the most common vulnerabilities. For example, the recent Mossack Fonseca firm notorious for Panama paper scandal was attacked by SQL Injection. Therefore it is necessary to examine the workings of these attacks and prevent or detect them. In this paper, the proposed system is classified into two modules active and passive. The proposed system tries to detect and prevent web applications from various attacks such as IP spoofing, MAC spoofing, SQL Injection, DDoS (Distributed Denial of Service) attacks.*

Keywords-SQL injection, IP spoofing, MAC spoofing, DDoS, Web applications, active module, passive module.

I. INTRODUCTION

People are using web applications for their everyday activities. Nowadays web applications are used for multiple purposes like for buying vegetables, clothes, electronic gadgets, etc. However they are also used for financial transactions, sharing sensitive data, etc. Web applications therefore handle a large amount of user's data. [1] As more and more of this sensitive and confidential data is being available on the net the interest of the attacker increases and it tries to reveal or steal this data. Web applications are basically a client-server mechanism where the server uses its database and other resources and tries to fulfill the requests made by the client. The

most dangerous and dominant web application attacks, exploit weaknesses in the system which are often associated with improper validation or mostly due to non-filtering of untrusted inputs, which results in the injection of malicious script or malicious code. [2]

Web Applications have become an integral part of the internet and most of the times these applications have sensitive and confidential data that has to be protected properly. So we need to keep the CIA that is confidentiality, integrity and availability of the application in mind to secure it. This paper studies different attacks which can create a threat to CIA of the web application. The paper also focuses on techniques to prevent and detect these attacks. The scope of the paper will be mostly based on IP and MAC spoofing, SQL injection and DDoS attacks mostly.

A. IP Spoofing

In IP Spoofing the attacker tries to impersonate another machine by changing its IP address. It's also called as IP address spoofing. It involves creating an IP packet (Internet Protocol packet) having a false IP address which is mostly used for unauthorized access. Here the attacker tries to impersonate the IP address of the legitimate user and tries to gain the access of the system using this IP address. It is one of the most widely used techniques to gain an unauthorized access or hiding identity for the attacker. IP spoofing is mostly used for DOS attacks that are denial of service attacks.

B. MAC spoofing

MAC spoofing is used to change the MAC address (Media access control address). The MAC address is factory-assigned address for the network interface of a network device. But still it can be easily changed. Attackers use this vulnerability for identity masking. It can be therefore used to hide the identity of the attacker.

C. SQL injection

SQL injection attack is the attack on the database of the system. It's a back-end attack where the attacker tries to insert or inject a SQL query from the input of the client side. A SQL injection attack can lead to leak of sensitive data or modification of data such as update-delete-insert operations in the data. An attacker can use this attack to impersonate that he is a legitimate user and bypass all security mechanisms and further he can use SQL injection for gaining entire control or access to the database. [1]

SQL Injection is one of the most popular and widely used attacks by the attacker. SQL injection attack is an attack on the Confidentiality of the data and Integrity of the web application.

D. DDoS

Distributed Denial of Service. It is an attempt used for making any online or network based service inaccessible to its users. It is done by momentarily or indefinitely disrupting services of the target machine or device connected to the net. In DDoS the target machine is flooded with requests by one or more machines so as to make the resources or services of the machine unavailable for the legitimate users. The network traffic at the target machine is increased by flooding it with requests so that the network becomes unavailable for legitimate users.

In this paper, we try to take look at various methods which have been suggested previously and try to merge most of them to form a secure web application. The section II in this paper discusses about the related work processed in the current topic. The proposed system is mentioned in Section III of this paper. Section IV states the conclusion and future work which needs to be processed to enhance the web application security.

II. RELATED WORK

Various research works are being conducted daily to address the weaknesses and web attacks in the web applications. [3] In the network or web application, various solutions for detection as well as prevention for Intrusion have been proposed. Their main objective is same but they differ in how they are implemented and what techniques they have utilized. On the basis of that, we have made the classification of the Intrusion Detection System in two primary modules:

(1) Host Level Intrusion,

(2) Network Level Intrusion.

Paper "Proceedings of the 7th European Symposium on research in Computer Security" has proposed an IDS that is based on anomalies in the network. This approach relies primarily on the data that the database writes to the host, which is also known as "trace data". Although, it does not focus on the prevention of the damage to the primary database which is the primary purpose of an IPS as previously discussed.

At the moment we have witnessed that a lot of work is going on web applications that is related to the security as well as IDS attached to it. One such method was proposed by Raghuveer and Chandrasekhar that includes the techniques of an SVMC (Support Vector Machine Classifier that classifies data based on SVM) along with K-means algorithm that creates k number of clusters from the given input as a dataset. These techniques are combined to the concept of a neural network in the fuzzy spectrum where they use the clusters to form trained datasets. [4]

One of the most common techniques used by an attacker that induces vulnerability in the system is SQL Injection. From the past twenty years decades, different researches and techniques have been presented and distributed by many researchers for detection and prevention of SQL Injection Attack (SQLIA). The top most priority in web application's security problems is SQLIA. By using simple and sophisticated commands attackers can modify or restructure the database. There was also a tool built in order for vulnerability and attack injection by Fonseca, Madeira and Vieira. This was done in order to evaluate the security of a web application completely, from all angles by the usage of a fault injection method. [5]

These security threats have different working mechanisms and hence, we try to incorporate all the security mechanisms for each one of the possible attack on vulnerabilities of the web client. These mechanisms are not all present in each of the websites on the internet; a website may contain few of these mechanisms and are vulnerable to those types of attacks that are either unstoppable or not discovered yet. By the careful analysis of the structure of a website a hacker may try to find out its vulnerability whose security implementation has not been done (yet) and may exploit that weakness.

III. PROPOSED SYSTEM

In this section we will introduce efficient methods for detecting web application attacks. The proposed system consists of two different modules based on their roles in the system. They are named as active module and passive module as per their functions.

a) Active module:

This is module is named active because in this module the system constantly tries to read the packets that it receives at the interface and takes decisions accordingly. This module actively reads out the packet data and compares the queries with the general defined rules to identify and attacks like SQL injections. The module reads the packet that arrives on the interface and further analyses only those packets that contain SQL queries.

The module filters the packets on the interface to evaluate the query frame in the packet. This is done by using a Python library Scapy. Scapy is a powerful packet manipulation python program that can read, create, decode and forge packets on the wire. The proposed module monitors the traffic. The application detects SQL Injection keywords from the traffic and blocks the malicious packets. [6]

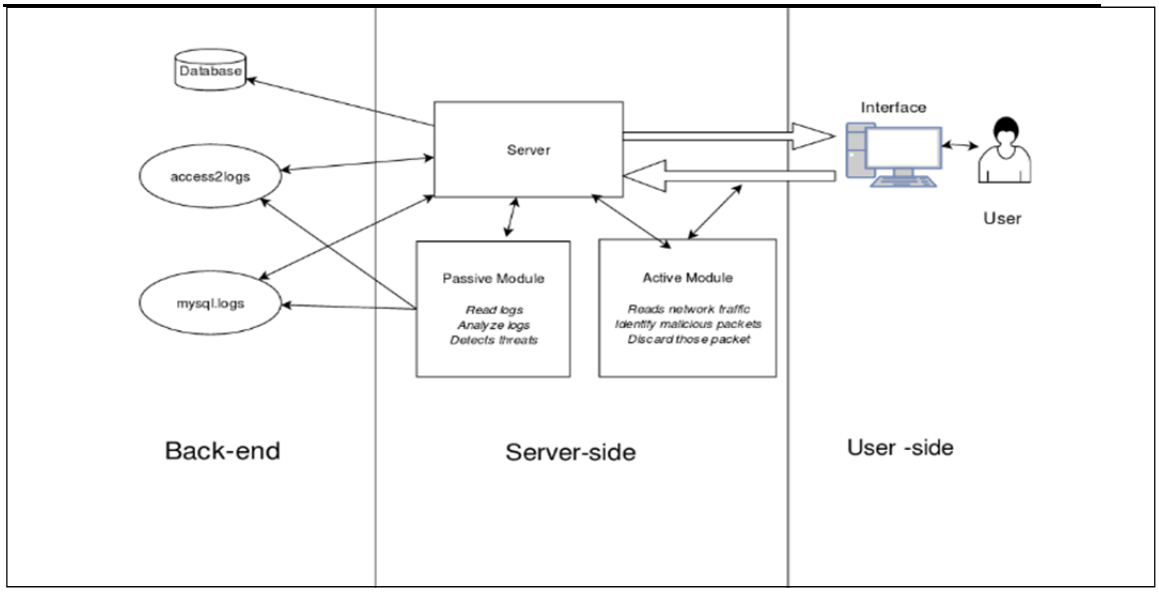

Fig 1(a). Proposed System block diagram

In a web application the user has a right to execute a very limited number of SQL queries. In most of the cases it is “SELECT element1, element2, from database”. These queries are basically data retrieval queries, data creation or data modification queries. The SQL query is formed by the input the user enters in the forms. The adversaries can enter the form input in such a way that the query formed turns into a malicious query and ends up compromising the database

An example for SQL injection attack is where the hacker carefully crafts a string that has a condition true for instance, this validates as True in database and the database in return will send the appropriate response with all the information related to that query.

Note that the user should not use in any case a query like “SELECT * from database” , because that would mean that the user wants to retrieve all the elements from the database. This is a red flag because the user wants to access elements that he is not permitted to access, thus flagging it as potential SQL injection attack.

For identifying potential threats that are masked into SQL queries, the module refers to set of keywords that it actively searches in the “INFO’ part in the packet. These keywords are set of common malicious SQL statements that are matched with the incoming data.

These keywords include Select, Delete, Insert, Update, Create, Drop, Alter, Where, Like, Or, ’=’, From. So if the info has query request which contains these keywords or multiple keywords then the application will simply discard the packet.

b) Passive module:

This module derives its name from its non-active nature. Every web application that is hosted on a web server has one overlooked feature that is the ability to every single activity of the web application server to log the processes. There are multiple logs that are created by the system and reading those logs and analysing them. The logs records everything that has happened with the server, understanding them will be give us the insight of what the server has been attacked with. [7]

Both the modules are running according to the administrator's configuration. It is recommended that the active module should always run actively analyzing the traffic.

A. SQL INJECTION PREVENTION

SQL injection the most common type of attack the web applications face. Even though it is the most common attack, millions of web applications are still unable to defend it. It is necessary to check to actively the traffic to see the potential attacks on the system. To do this, Each packet has the following attributes, as shown in Fig 1(b).

No.	Time	Source	Destination	Protocol	Length	Frame length on the wire	Query	Info

Fig 1(b). Packet Attributes

The module filters the traffic searching for SQL queries in the packets. In the SQL query the user has rights to execute are very limited in any web application. In most of the cases it is “SELECT element1, element2, from database”. Note that the user should not use in any case a query like “SELECT * from database”, because that would mean that the user wants to retrieve all the elements from the database. This is a red flag because the user wants to access elements that he is not permitted to access thus flagging it as potential SQL injection attack.

For identifying potential threats that are masked into SQL queries, the module refers to set of keywords that it actively searches in the “INFO’ part in the packet. These keywords are set of common malicious SQL statements that are matched with the incoming data.

Passively the system reads the logs of the server. The logs that the system specifically reads are the access logs. These logs are created by the server for each http request the user sends to the server. The logs are universally in the format of:

“IP address”, ”Timestamp”, ”Request from client”, ”status code”, ”size of the object returned”

The passive module maps the request from the client, status code and size of the object returned with respect to its IP address. This is done using regular expressions. Each line of the log is parsed with the regular expression and converted into a data structure. This data structure is stored in a file periodically. If the Request from client contains unusual elements like “POST” and ”DELETE” then the IP that made the request will be blocked using IP tables.

The status codes are the one of the key features of the client server architecture. These status codes are as follows:

- 1xx Informational responses
- 2xx Success
- 3xx Redirection
- 4xx Client errors
- 5xx Server error

The passive module actively looks out for status code 200 that is the OK message send by the server to the client. Any other code that is present in the logs will be created as an alert to the administrator.

B. IP AND MAC SPOOFING PREVENTION

The system maintains a database of each IP address and MAC address pair that accesses the system. There can be a scenario where one IP address will have multiple MAC addresses that is usage of public IP address. In this scenario multiple IP addresses are mapped to one MAC address. But the system does not allow more than two MAC addresses belonging to the same IP address. This is to prevent from MAC spoofing as well as IP spoofing. The web application does not accept any IP address that originates from outside its network. This ensures that the web application is not accessed by IP addresses outside its own network. The system blocks the IP address if they share the same MAC address.

IV. CONCLUSION AND FUTURE WORK

In this section we are going to conclude this paper as well as enlist the main steps we are going to carry out in the future work for the project.

A system is proposed which is based on integrating various methods of preventing attacks on web applications. The methods implemented are SQL query analysis for preventing SQL injections, Parsing and analysing web server logs to analyse and defend against unauthorized access to the system, bad requests. Our system can also prevent attacks like Denial of service by setting up the IP tables to drop the suspicious packets. By applying the security rules to the system, we will be able to defend the system against attacks like MAC address spoofing and IP spoofing as well. These security rules will be determined by the administrator.

In Future, the system can be improved to further to optimize the query analysis using search algorithms. More modules are to be added, specifically focusing on another common attack Cross site scripting. Another module will be created to perform integrity check on server files to prevent tampering of sensitive and important server files. This will be implemented by creating the hash values of those files and rechecking the hash periodically. Since most websites create content on the client machines, the server files should be tampered most of the times.

V. REFERENCES

[1] Piyush A. Sonewar, Nalini A. Mhetre, "A Novel Approach for Detection of SQL Injection and Cross Site Scripting Attacks", International Conference on Pervasive Computing, 2015.
[2] Dafydd Stuttard, Marcus Pinto, "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2nd Edition", September 2011.
[3] R. V. Bhor and H. K. Khanuja, “Analysis of web application security mechanism and attack detection using vulnerability injection technique”, Department of Computer Engineering, MMCOE, Pune, India, 2016.
[4] A. M. Chandrasekhar, K. Raghuveer “Intrusion Detection Technique by using K-means, Fuzzy Neural Network and SVM classifiers”, 2013 International Conference on Computer and Informatics(ICCCI),Coimbatore, INDIA, Jan04-06,2013.
[5] J. Fonseca, M. Vieira, and H. Madeira, “Evaluation of web security mechanisms using vulnerability and attack injection,” IEEE Transaction on Dependableand Secure Computing , Vol. 11, Oct. 2014.
[6] Venkatramulu Sunkari, Dr.C.V.Guru Rao, “Preventing Input Type Validation Vulnerabilities Using Network Based Intrusion Detection Systems”, International Conference on Contemporary Computing and Informatics, 2014.
[7] MANJU KHARI, SONAM, VAISHALI, MANOJ KUMAR, “Comprehensive Study of Web Application Attacks and Classification”, Dept. of Computer Science & Engineering, Guru Gobind Singh Indraprastha University Delhi, India, 2016.