Component Based Web Application Firewall for Analyzing and Defending SQL Injection Attack Vectors

Prabhat Bisht, Manmohan Singh Rauthan, Raj Kishore Bisht

Abstract: Structured query language injection is a top rated vulnerability by open web application security project community. If a web application has structured query language vulnerability in source code, then such application is prone to cyber-attacks, leading to attack on confidentiality, integrity and availability. Attackers are always ready to exploit structured query language injection vulnerabilities by executing various online attack vectors and many times successfully bypass authentication and authorization to gain privilege access on web and database server leading to service interruption, data interception, modification, fabrication and sometime complete deletion of database. The present paper is an attempt to propose an advance component based web application firewall to enhance web application security by mitigating structured query language injection attack vectors by analyzing hypertext transfer protocol request variables through analyzer component and defending injection attack through defender component based on content policy installed on advance web application firewall.

***Keywords**: Advance Web App Firewall (AWAF), Hypertext Transfer Protocol (HTTP) Open Web Application Security Project (OWASP), Structured Query Language injection (SQLi)*

I. INTRODUCTION

Web application is a line of codes written in web programming language deployed on web server hosted in cloud enabled datacenter for delivering services over internet. If such applications are not properly coded in terms of security then it left vulnerabilities in its source code, later on attackers exploit such vulnerabilities. Open web application security project (OWASP) [1] community identified top 10 vulnerabilities which are highly exploited by attackers. These vulnerabilities are SQL injection, broken authentication and authorization, cross site scripting (XSS), broken access control, XML External entities, sensitive data exposure, component with known vulnerabilities, insecure application programming interface known as API , application and server level security misconfiguration and insecure deserialization.

We have surveyed last few year records related to major security breaches globally with the emerging of cloud based datacenter from year 2012 and find that 70% security breaches are because of only SQL related vulnerabilities in web application source code, where attackers performed injection attacks leading to valuable data loss and compromise database security and privacy.

In year July 2012 [2] a group of attackers popularly known as "D33Ds Company" identified vulnerabilities in yahoomail.com and performed union based SQL injection attack and publicly posted password of 450,000 Yahoo users, it was the massive loss for Yahoo users and Yahoo.

Year 2014 [3] was dominated by cyber-attacks, and most of the attacks were related to injection attack vectors. In January 2014, credit card details of 20 million South Korean were stolen, same year Microsoft Blog, CNN-website, Twitter and Facebook hijacked by Syrian electronic army. Many more such attacks were performed throughout the year. SQL Injection was the base of most of the attacks.

In year 2015[4] a report was published claiming that SQL injection is a top performing attack of year 2015 and most of the attacks were carried out in government and financial sectors . So many similar attacks were performed throughout the year 2016 and 2017.

In February 2018 [5] there was a huge security breach in Heartland Payment Systems where attackers detect and exploit vulnerabilities in there online payment system and through SQL injection attack gained privilege access in their payment database. Attackers claimed that they have hacked personal and financial information of 160 million card users. It is one of the biggest security breaches in card data privacy.

On 13th May 2019 [6] a report was published in economic times claiming that 69% Indian firms and 63% of Australian companies are prone to cyber-attacks and that's because of vulnerabilities in their application software. Facepoint, a leader in global cyber security pointed that 83% of organizations of Asia Pacific Region never thought about security aspects related to digital transaction leading to financial data loss in case of attacks.

AppSec report [7] published in June 2019 is a reason for concern in cyber world as report pointed out that there are 53% increase in attack vectors in comparison to May 2019. Out of 53%, 39% of the attacks are related to unpatched vulnerabilities within application source code.

Manuscript published on 30 September 2019
* Correspondence Author

Prabhat Bisht*, PhD Scholar, Uttarakhand Technical University, Dehradun, India. Email: prabhatbisht@gmail.com

Dr. Manmohan Singh Rauthan, computer science and engineering, Hemwati Nandan Bahuguna Garhwal University , Srinagar Garhwal, Uttarakhand , India. Email: mms_rauthan@rediffmail.com

Dr. Raj Kishore Bisht, Bisht Mansion, Prem Vihar, Pilikothi, Haldwani Nainital Uttarakhand, India, Email: bishtrk@gmail.com

Alarming Observations of APPSEC Report

Rise in attacks on web applications increases 53% in June 2019 in comparison of May 2019. 39% of attacks are related to web vulnerabilities as identified by OWASP community. Less than 1% of attacks observed are related to open source components used in application development. The remaining attacks are related to cross site scripting, path traversal, session misconfiguration and mostly are of SQL injections.

Report shows that exploitation of vulnerabilities related to SQL injection is a top choice for attackers and 58% attacks are related to SQL injection only.

At the time of writing this paper in July 2019 [8] there is a massive cyber-attack in Bulgarian revenue agency where attackers hacked 5 million Bulgarians data by exploiting vulnerabilities in their application source code, claimed by attackers, it is noted that the total population of Bulgarian is 7 million, means that entire nation financial data is compromised with such attack.

After doing all this analysis, finally we have come to this conclusion that 2/3rd of attacks are of SQL injection. Attackers inject malicious code in SQL queries by intercepting HTTP request variables through proxy based tool like Burpsuit in order to find vulnerable web page after identifying vulnerabilities in web page perform attack. One of a reason for increasing attacks on web applications is emerging cloud technology. With the rise of cloud enabled data centers and advance web technologies, most of the companies are able to deploy websites over cloud platform knowingly or unknowingly without following web application security audit process.

It is found that cloud security is limited only up to network layer by network firewalls, cloud service providers provides infrastructure as a service, platform as a service. It does not provide security as a service in layer 4 application layer through some advance web application firewall for analyzing and defending web application vulnerabilities.

Here comes the role of advance layer 4 component based web application firewall in application layer of TCP/IP protocol as proposed in this paper for mitigating SQL injection and cross site scripting attack by analyzing HTTP request parameters and reject malicious request by analyzing request parameters based on security content policy deployed on advance web application firewall.

II. STATEMENT OF PROBLEM

In current scenario most of the organizations are hosting their web applications in cloud computing [9] infrastructure without knowing the security aspects associated with cloud platform , as it is economic, easy to host and works on pay as per usage policy. In data centers security is limited to network layer firewalls only (IAAS) .Cloud service providers does not provides adequate security for mitigating OWASP vulnerabilities in application layer.

As a result security threat landscape has transformed drastically from virus , worms , Trojan horses, denial of services (DOS) attacks to ransom ware, phishing , website hacking to distributed denial of services (DDOS) [10] attack and so on , there is no stoppage.

With the emergence of web 2.0 [11] and Cloud technology [12], information sharing in internet has increased tremendously over the past few years. There is dynamic growth in information sharing because of various social networking sites, use of financial related services through various e-Service delivery web portals and with adoption of web based business solution for doing business and delivering customer relationship services. Applications holding personal and financial related data are often attacked directly by attackers. Attackers either try to compromise the organization network or exploit the vulnerabilities of application software.

As discussed reports shows that more than 60% web attacks are because of SQL injection which typically results from flawed coding, failure to sanitize HTTP input variables and not concerning about writing secure code from the very first step of software development life cycle.

The purpose of this paper is to do literature review of various researches in the field of mitigating SQL injection attack vectors and to propose an advance component based web application firewall as security as service in cloud infrastructure for mitigating attack vectors.

III. LITERATURE REVIEW

Most of the research work has been carried in the field of web vulnerabilities. Open web application security project community is working in web application security guidelines and community identified Top 10 vulnerabilities which are highly exploited. These vulnerabilities are SQL injection flaws, cross site scripting (XSS), session management, broken authentication and authorization, broken access control mechanism, security misconfiguration, sensitive data exposure, cross site request forgery, using components with known vulnerabilities and under protected application programming interfaces (API).

These vulnerabilities are highly exploited by attackers in order to gain access to web server or web applications. Most exploited one is SQL injection .For gaining access to web server or web applications attackers intercept HTTP request parameters through proxy based tool like burp suite and inject malicious code in request variables. Malicious code when processed by web server results injection attack. By performing injection attack attackers gain privilege access to web server by passing authentication and authorization leading to attack on confidentiality, integrity and availability popularly known as CIA threat [13].

An intense literature review is done based on paper published in journals and conferences. Our study identify that most of the vulnerabilities are related to:

***Injection*:** Injection flaws such as SQL injection, Union based injection, Boolean based injection.

All these injection attack occurs when malicious code is injected in SQL query string and processed by web server. There is lacking of some advance filter which passes legitimate requests to web server and reject malicious one.

Research is going on regarding analyzing and defending online SQL injection attack vectors.

In past, Vieira, Fonseca and Madeira [14] had developed vulnerability and attack injection tool for an evaluation of web security mechanism using fault injection technique.

Retrieval Number C4674098319/2019©BEIESP
DOI: 10.35940/ijrte.C4674.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Haibin Hu [15] proposed various characteristic and procedure for analyzing SQL injection attack. In his study he proposed defense resistance and remedy model from the perspective of non-intrusive SQLI attacks. Subhranil Som [16], Sapna Sinha, Ritu Kataria proposed a two phase security model for SQLI attack in stored procedure.

Asish Kumar Dalai [17] and Sanjay Kumar Jena proposed a model for classification of SQL injection vulnerabilities by creating a dataset and used proxy based tools for injecting malicious code in request parameters for analyzing and defending SQLI attack.

Sonakshi [18], Rakesh Kumar, Girdhar Gopal in his paper proposed case study of various SQL injection attack vector performed live on websites.

Hossain Shahriar [19] Sarah North and Wei-Chuen Chen present a client-side approach to analyze and detect SQL injection attacks. Their study is based on the concept of shadow queries for detection of SQL injection in early stage in client side only. Browser i.e. client receive shadow queries from input as supplied from server and checks deviation in between shadow queries and dynamic queries. The measure of deviation is calculated based on entropy metrics and propose four metrics in this direction for early detection of SQL injection attack. Zoran Djuric [20] had developed a SQL injection vulnerability detection tool using black box approach to analyze HTTP request and response from the web server. Anh Nguyen-Tuong [21] proposed a fully automated approach for securely hardening web applications. It checks for tainted data and dangerous content came from untrustworthy sources.

Hsiu-Chuan Huang [22] proposed a tool VULSCAN, this scanner automatically generates test data for revealing vulnerabilities in source code.

Deepak Dattatray [23] proposed a cloud based system to sense security vulnerabilities of web application in open source private cloud infrastructure as a service (IAAS).

Sajjad Rafique , Mamoona Humayun, Zartasha Gul, Ansar Abbas, Hasan Javed [24] conducted systematic literature review to investigate different vulnerabilities in web application from the very first beginning of software development life cycle.

Prabhat Bisht [25], Devesh Pant, Manmohan Singh Rauthan in his paper proposed an advance security model known SECUREWEB for analyzing and defending web application vulnerabilities in cloud computing. It works as advance source code scanner which sense vulnerabilities in web application source code and patch them automatically.

IV.RESEARCH METHODOLOGY

It is clear that cyber-attacks are increasing day by day and there are two main reasons behind it.

***Missing Application level security*:** OWASP community is working in this direction and time to time they are updating vulnerabilities list. These vulnerabilities must be avoided at the very first step of software development life cycle.

Missing web server level security in cloud: There is a need of some advance web application firewall in cloud for mitigating SQL Injection and related attack vectors, as SQL injection is one of a top performed attack. Most of the available web application firewalls are limited to traffic analysis only not application source code content analysis.

Current research is based on analyzing different formats of SQL injection attack vectors by extracting keywords used by attackers in SQL injection attack vectors and thereafter formulation of malicious injection attacks content policy based on keywords extracted from malicious HTTP request parameters and thereafter developing advance web application firewall for filtration of malicious and legitimate requests based on content policy.

A. Definition of injection attack

Injection attack is a technique in which attacker insert arbitrary SQL commands in the queries that a web application executes to its database through vulnerable web pages. Attackers inject malicious codes in the request parameter through proxy based tools ex. Burpsuit and pass it to web server through vulnerable web page leading to execute such code in backend database like MySQL, Oracle, and PostgreSQL. A successful attack can lead to unauthorized access to sensitive information like personal and financial information by bypassing authentication and authorization leading to huge data loss or sometime complete shut down or even deletion of whole database.

B. Types of SQL injection

- *Error based simple SQL injection*

This is most common SQLi attack performed by attackers, attackers insert unexpected commands i.e. invalid input in request parameters as man in the middle attack by intercepting request variables through some tools. Such commands when executed in web server return unexpected response as a form of error and display database structure, table structure, database version and sometime returns full query in web browser. Such information later exploited by attackers.

Demonstration of error based SQL injection attack

***Client side** :* web page
<form name=”myform” action=”boolean.php” >
<input type =”text” name=”forward_request” value=”<?php echo $someid;?>” />
<input type=”submit” name=”submit1” value=”add”>
</form>
Server side: Vulnerable SQL query in boolean.php.
SELECT column from table where column_id =’$someid’;
Payload: Attacker intercept request variables and inject malicious code as man in the middle attack.
CurYear=2019&CurMonth=08&CurDay=13&**someid=5’** +&submit1=add
Response: Injected code get executed on web server.
SELECT column from table where column_id =’5’’**;**
[note the extra code here, injected by attacker]
Query executed and server return unexpected response.
SQL>SELECT column from table where column_id =’5’’;
ERROR 1064 (42000): You have an error in your SQL syntax ;check the manual that corresponds to your SQL server version for the right syntax to use near ‘5’’ at line 1

- *Union based SQL injection*

In union based SQL injection attack, attacker use two SQL queries using UNION operator in between for performing SQL injection attack vector. The information returned by web server after executing such malicious SQL statements is exploited by attackers to gain knowledge about database structure and table's structure to perform attack vector. There are two prerequisite before performing this attack. First, each SQL statement within UNION has same number of columns and in same order. Second, columns in each statement have similar data type.

Demonstration of union based SQL injection attack vector:

Client side: HTTP request from client browser.
http://localhost:8080/vulsite/unionattack.php/?id=1%20+ UNION+SELECT+1%3B--+&Submit=SUBMIT#

Server side: Union based SQL query in unionattack.php.
SELECT fname, lname from users UNION username, password from login;

Payload 1: Attacker performed union based SQL injection attack in between two tables. If tables having same no of columns and same type of data types in same order then injected query return output as desired by attackers.
SQL> SELECT fname, lname from users WHERE userid='1' UNION SELECT 1,2;--';

Response : Query returns expected output for exploitation.

```
fname        |  lname
Admin        |  admin
      1         |  2
```

Payload 2: Performed union based SQL injection in between two tables, tables having unequal no of columns and different data types returns unexpected output.
SQL> SELECT fname, lname from users WHERE userid='1' UNION SELECT 1,2,3;--';

Response: ERROR 1223 (2100): incorrect data type or columns. Such information is exploited by attackers

An attacker can test multiple variants of UNION based SQL injection with different combinations until they hit the right one. Further exploiting these vulnerabilities attackers can obtain name of a table along with columns name and later perform attack vector to exploit database.

- *Boolean based Blind SQL injection.*

This type of injection attack does not show any error message, hence named blind. This injection attack returns only true and false response from web server. By observing response, an attacker can extract sensitive information.

Demonstration of Boolean based blind SQL attack vector:

Client side: HTTP request from client browser.
http://localhost:8080/vulsite/booleanattack.php/?id=1&Submit=SUBMIT#

Server side: booleanattack.php [Vulnerable web page]

```
<?php  $id =$_GET['id'];
$sql="select fname,lname from users where userid=
'$id'";
$result=mysqli_query($sql);$num=@mysqli_num_rows(
$result);
If($num >0)
 $html=" user id exists in database";
else
 $html="user id does not exist in database";
?>
```

- *Time-based Blind SQL injection*

Base of time based SQL injection is delay tactics, means if a web application is vulnerable to SQL injection then it will wait for a specific time before a vulnerable web page respond for attackers queries. Attackers insert time delay value using sleep () function as an attack payload.

Demonstration of time based blind SQL injection:

Client side: *HTTP request from browser.*
http://localhost:8080/vulsite/timebooleanattack.php ?id=1'+and +sleep(10);--+&Submit=SUBMIT#

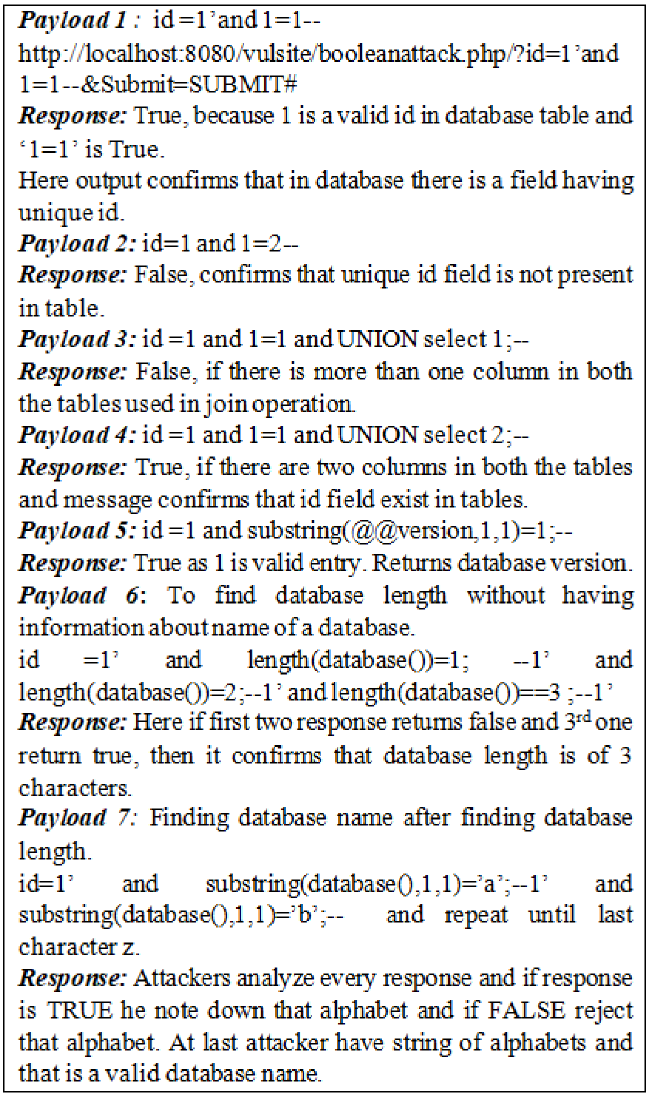
Server side :

Payload 1 : id =1 and sleep(10);--

Response: If above web page is vulnerable then response comes after 10 second.

Payload 2: id =1 and if ((select +@@version) like "10",sleep(5),null);--+

Response: If response comes within 5 seconds then it confirms that database version is 10.



Payload 1 : id =1'and 1=1--
http://localhost:8080/vulsite/booleanattack.php/?id=1'and 1=1--&Submit=SUBMIT#

Response: True, because 1 is a valid id in database table and '1=1' is True.
Here output confirms that in database there is a field having unique id.

Payload 2: id=1 and 1=2--

Response: False, confirms that unique id field is not present in table.

Payload 3: id =1 and 1=1 and UNION select 1;--

Response: False, if there is more than one column in both the tables used in join operation.

Payload 4: id =1 and 1=1 and UNION select 2;--

Response: True, if there are two columns in both the tables and message confirms that id field exist in tables.

Payload 5: id =1 and substring(@@version,1,1)=1;--

Response: True as 1 is valid entry. Returns database version.

Payload 6: To find database length without having information about name of a database.
id =1' and length(database())=1; --1' and length(database())=2;--1' and length(database())==3 ;--1'

Response: Here if first two response returns false and 3rd one return true, then it confirms that database length is of 3 characters.

Payload 7: Finding database name after finding database length.
id=1' and substring(database(),1,1)='a';--1' and substring(database(),1,1)='b';-- and repeat until last character z.

Response: Attackers analyze every response and if response is TRUE he note down that alphabet and if FALSE reject that alphabet. At last attacker have string of alphabets and that is a valid database name.

C. Diagrammatic representation of SQL injection attack vectors.

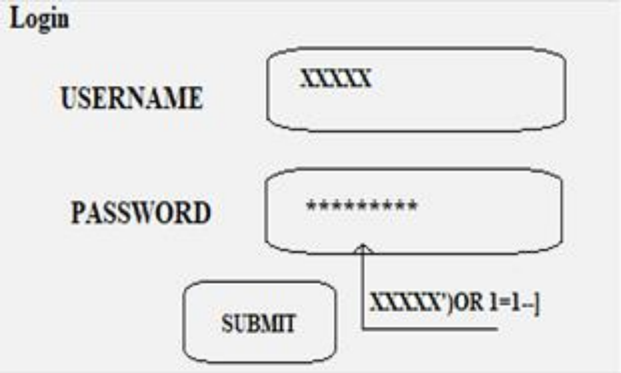


Fig. 1. Attack payload performed in login form

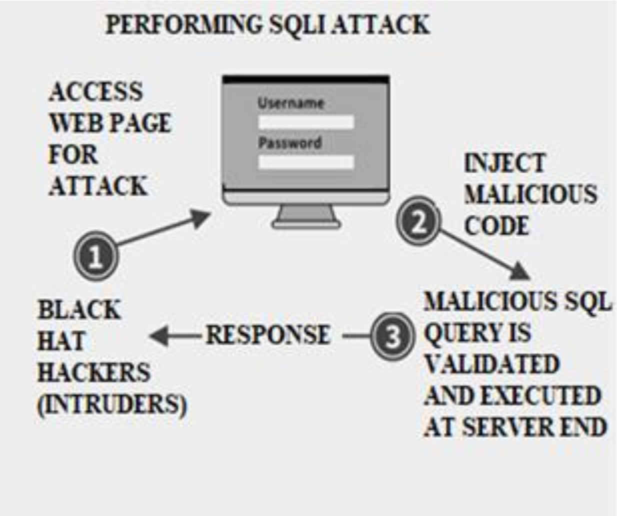

PERFORMING SQLI ATTACK
ACCESS WEB PAGE FOR ATTACK
Username
Password
1
INJECT MALICIOUS CODE
2
BLACK HAT HACKERS (INTRUDERS)
RESPONSE
3
MALICIOUS SQL QUERY IS VALIDATED AND EXECUTED AT SERVER END

Fig. 2. Attack vector.

D.Formulation of content policy.

After studying various SQL injections attack vectors we have identified certain key parameters here we call contents through which attackers try to exploit vulnerabilities in web pages. If such attack vectors get rejected by web application firewall by content analysis then attackers can never gain access in application and database server and it will be a great achievement in the field of application security.

Table- I. Content policy RESULTSET

SQL injection	Query contents /attack vector keywords
Error based	‘ i.e. extra quote in SQL statement
	SELECT,CONCAT,@@VERSION Avoid all special characters in SQL statement , =!@#$%^&*()+-_={}[]:;”’
Union based	‘ ,UNION SELECT ;-- @@version Ex. id =1’ UNION SELECT 1;--
Boolean based Blind SQL	’, and ,union ,select ,;-- ,substring, length Ex. 1’ and 1=1 UNION select 1;-- Ex. Substring(@@version,2,1)=1;-- Ex. length(database())==3;--
Time based blind SQL	Sleep();-- Ex. 1’ and sleep(10);--

V. PROPOSED ARCHITECURE

Here we are describing architecture of proposed advance web application firewall which works in layer 4 of TCP/IP protocol in application layer.

In our study we observed that most of the security is limited at layer 2 in network layer of TCP/IP protocol through implementation of network firewall that filter incoming and outgoing traffic based on predetermined set of rules implemented as per organization policy. Network firewalls works as a barrier between trusted and non-trusted network. But it does not provide security to application software deployed as software as a service in cloud datacenters.

Currently few cloud service providers are providing security as a service through layer 4 firewalls but such firewalls are limited to blocking and monitoring input, output or certain services as per configured policy that is limited to traffic analysis only, not providing adequate security as a service for migrating OWASP Related vulnerabilities especially SQLi attacks by content analysis.

So here we proposed advance web application firewall having two components.

- ***Analyzer :*** it is a first defense system, REQUEST first passes through analyzer component , analyzer detect vulnerabilities in request variables , it look up the content policy installed over it and check whether there is any pattern available in request variables which is available in content policy. If pattern matches then request forwarded to defender component for further action
- ***Defender:*** defender thoroughly analyzes the content and classifies injected malicious code as SQL injection attack or some other OWASP attack. If the attack is of SQL injection then defender completely blocks that IP from where request is generated. Update logs and alert application administrator by SMS or email. If the attack is of other then SQL injection then defender sanitize request variables with input filters and thereafter forward it to web server for processing. Sanitization modifies input to ensure that is safe for processing.

Web application firewall works as a defense system for mitigating SQL injection attack and other OWASP vulnerabilities by inspecting request variables at application layer based on content policy as suggested and deployed on web application firewall. Time to time content policy should be updated so that attacker cannot find a way to bypass web application firewall.

Besides implementing network layer firewall and application layer firewall, some precautionary measures should be adopted at the very first step of software development life cycle like.

- Validating user input at client and server side.
- Use of prepared statement for performing queries, in this way attacker cannot modify backend queries by injecting malicious code.
- Proper authentication and authorization at application and database level, timely security auditing and patch management is necessary for web security.

V1. FLOWCHART OF PROPOSED ADVANCE WEB APPLICATION FIREWALL

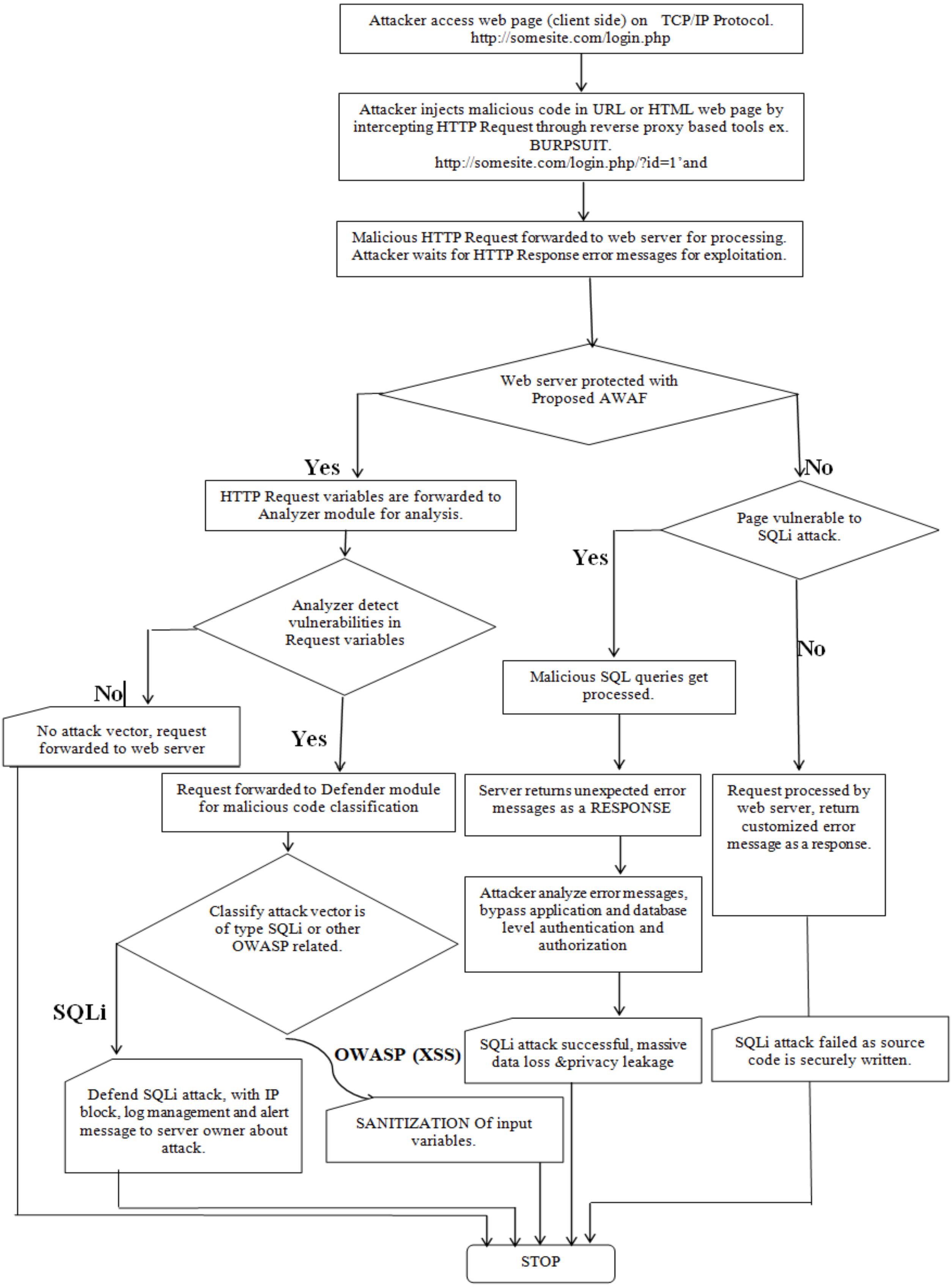

Attacker access web page (client side) on TCP/IP Protocol. http://somesite.com/login.php
Attacker injects malicious code in URL or HTML web page by intercepting HTTP Request through reverse proxy based tools ex. BURPSUIT. http://somesite.com/login.php/?id=1'and
Malicious HTTP Request forwarded to web server for processing. Attacker waits for HTTP Response error messages for exploitation.
Web server protected with Proposed AWAF
Yes
No
HTTP Request variables are forwarded to Analyzer module for analysis.
Page vulnerable to SQLi attack.
Yes
Analyzer detect vulnerabilities in Request variables
No
Malicious SQL queries get processed.
No
No attack vector, request forwarded to web server
Yes
Request forwarded to Defender module for malicious code classification
Server returns unexpected error messages as a RESPONSE
Request processed by web server, return customized error message as a response.
Classify attack vector is of type SQLi or other OWASP related.
Attacker analyze error messages, bypass application and database level authentication and authorization
SQLi
OWASP (XSS)
SQLi attack successful, massive data loss &privacy leakage
SQLi attack failed as source code is securely written.
Defend SQLi attack, with IP block, log management and alert message to server owner about attack.
SANITIZATION Of input variables.
STOP

Fig . 3. Flowchart of advance WAF.

VII. ARCHITECTURE OF PROPOSED WEB APPLICATION FIREWALL IN CLOUD

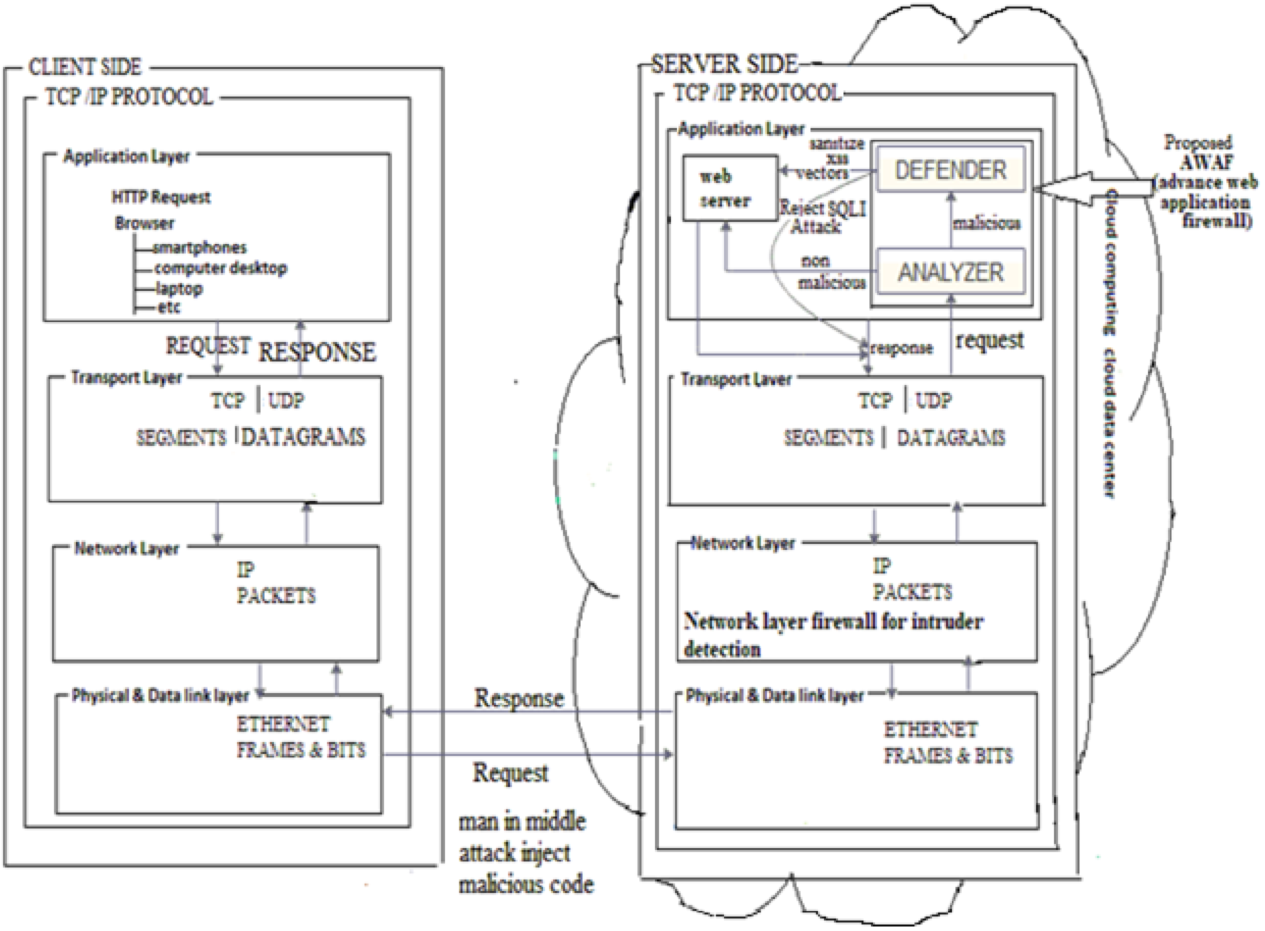


Fig.4. Deployment of proposed component based web application firewall in cloud computing works in TCP/IP Model.

CONCLUSION

During this research, we find that cyber-attacks are growing day by day impacting our social economic and personal life and in greater extent threat to one country security. Our valuable data is not secure in cloud as attackers are always ready to find vulnerabilities in web based applications by inventing and performing new attack vectors. Most of the mission mode web applications of government and corporate sector having data related to national security and financial are always in radar of attackers, attackers try to find vulnerabilities in such applications by inventing new attack vectors leading to cyber terror. In this work we analyzed different type of SQL injection attack vectors and proposed an advance layer four web application firewall which works in application layer of TCP/IP protocol and deployed it in cloud enabled data centers to safeguard web applications from OWASP attack vectors. This research work can prove to be a benchmark in application security by proposing an advance web application firewall based on analyzing various SQL injection attack vectors and through identification and formulation of vulnerable content policy as a result for mitigating various online SQL injections and cross site scripting attack which are top rated attack vectors.

REFERENCES

1. https://www.owasp.org [Accessed 18-6-2019]
2. Kerner, Sean Michael. (2012, july12). Yahoo password breach puts SQL injection in the crosshairs. https://www.esecurityplanet.com
3. Morgan, Lewis. (2014, December 23rd). List of cyber-attacks and data breaches in 2014. https://www.itgovernance.co.uk/
4. Bradley, Nichlos. (2015). IBM 2015 cyber security intelligence index. https://essextec.com/
5. Lewis,Dave .(2015, May 31st) . Heartland Payment systems suffers data breach. https://forbes.com
6. IANS. (2019,May 13) .69% of indian firms face serious cyber-attack risk : study .https://economictimes.indiatimes.com
7. Watson,Katharine .(2019,july 19) . june 2019 AppSec intelligence report attack addition .https://www.contrastsecurity.com
8. Rothwell,James .(2019,July 16) .Attackers steal personel details of two in three Bulgarian as they mock joke cyber security .https://www.telegraph.co.uk/news
9. A. Srinivasan and J.Suresh, "Cloud Computing A practicle approach for learning and implementaion," *Pearson, 978-81-317-7651-3, 2014.*
10. Kramer L," Monitoring and Defending Against Amplification DDoS Attacks. In: Bos H., Monrose F., Blanc G. (eds) Research in Attacks, Intrusions, and Defenses. RAID 2015. Lecture Notes in Computer Science, vol 9404," *Springer*
11. Paroutis, S. and Al Saleh, A. (2009), "Determinants of knowledge sharing using Web 2.0 technologies," *Journal of Knowledge Management Vol. 13 No. 4, pp. 52-63. https://doi.org/10.1108/13673270910971824*
12. S. Kumar and R. H. Goudar, "Cloud Computing – Research Issues, Challenges, Architecture, Platforms and Applications: A Survey," *International Journal of Future Computer and Communication, Vol. 1, No. 4,pp. 356360, December 2012 .*
13. samonas, spyridon, coss, david, "the cia strikes back: redefining confidentiality, integrity and availability in security, " *journal of information system security . 2014, vol. 10 issue 3, p21-45. 25p."*

14. Jose Fonseca, Marco Vieira, Henrique Madeira, "Evaluation of Web Security Mechanisms Using Vulnerability & Attack Injection," *IEEE Transactions On Dependable And Secure Computing, Vol. 11, No. 5, September/October 2014*
15. Haibin Hu (2017), "Research on the Technology of Detecting the SQL Injection Attack and Non-Intrusive Prevention in WEB System," *AIP Conference Proceedings 1839, 020205 (2017)*
16. Subhranil Som ,Sapna Sinha ,Ritu Kataria," Study On Sql Injection Attacks: Mode, Detection And Prevention," *International Journal of Engineering Applied Sciences and Technology, 2016 Vol. 1, Issue 8, ISSN No. 2455-2143, Pages 23-29* Published Online June - July 2016 in IJEAST (http://www.ijeast.com)
17. Asish Kumar Dalai and Sanjay Kumar Jena," Neutralizing SQL Injection Attack Using Server Side Code Modification in Web Applications", *Hindawi Security and Communication Networks Volume 2017* Article ID 3825373, 12 pages https://doi.org/10.1155/2017/3825373
18. Sonakshi, Rakesh Kumar, Girdhar Gopal," Case study of SQL injection attack," *international journal of sciences and research technology 2016 ISSN: 2277-9655*
19. Hossain Shahriar, Sarah North, and Wei-Chuen Chen," Early detection of SQL injection attacks,"*International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.4, July 2013*
20. Zoran Djuric, "A Black-Box Testing Tool for Detection of SQL Injection Vulnerabilities," *ISBN: 978-1-4673-5256-7/13*
21. Anh Nguyen-Tuong, Salvatore Guarneri, Doug Greene, Jeff Shirley, David Evans, "Automatically Hardening Web Applications Using Precise Tainting," Department of Computer Science, University of Virginia, 151 Engineer's Way, Charlottesville, VA 22904-4740, USA
22. Hsiu-Chuan Huang, "Web application Security: threats Countermeasures, and pitfall," *COMPUTER Volume 0018-9162/17 @ 2017 IEEE*
23. Deepak Dattatray,"A Cloud based system to sense security vulnerabilities of web application," 2016 *International conference (ICEECCOT)*
24. Sajjad Rafique, Mamoona Humayun, Zartasha Gul, Ansar Abbas, Hasan Javed,"Web Application Security Vulnerabilities Detection Approaches : A Systematic Mapping Study," *2015 IEEE/ACIS 16th international conference*.
25. Prabhat Bisht, Devesh Pant, Manmohan Singh Rauthan "analyzing and defending web application vulnerabilities through proposed security model, " *International journal of science and technology. Vol. 6, Issue 2, 183-196, 2018 ISSN: 0975-1416 (Print), 2456-4281 (Online)*

AUTHORS PROFILE

Prabhat Bisht is currently pursuing PhD in Computer Science and Engineering from Uttarakhand Technical University Dehradun. His area of interest is Cloud Computing, Cyber Security, Application Security and Machine Learning. He has presented research paper in international conference of recent innovation in electrical electronics and communication system (RIEECS) and published paper in international journal of science and technology.

Dr Manmohan Singh Rauthan is full time professor in Hemwati Nandan Bahuguna Garhwal University, Uttarakhand in department of computer science and engineering , his area of interest is Cloud Computing, Natural Language Processing, he has several publication is reputed journals. He has vast academic experience. He guided many PhD scholars and highly dynamic person.

Dr Raj Kishore Bisht is associate professor in one of reputed engineering college of Uttarakhand; he has completed his PhD in Topic: Natural language processing through different mathematical and statistical tools. He has several publications in his name. He has guided several research scholars highly proficient in mathematics. Highly motivated and dynamic person.