Defending Against Web Application Cross-Site Scripting Attacks (XSS) Using new security model (NSM)

Dr. Khalid Ahmed Ibrahim Khalifa

Associate Professor, Karary University, Khartoum, Sudan.

Dr. Yousif Abd Elmalik Gasm Elseed Mohamed

Assistant Professor, Delat College of science and technology,
Khartoum, Sudan.
Y_abdalmalik@yahoo.com

المستخلص:

ان توفير الأمن لتطبيقات الويب ضد الهجمات السيبرانية في السنوات الأخيرة يعتبر هاجسا للعديد من الشركات والمستخدمين، وتأتي في مقدمة هذه التهديدات هجمات حقن SQL وتحتل هجمات البرمجة النصية عبر المواقع(XSS) وهي نوع خاص من هجمات الحقن الأولوية القصوى على تهديدات المواقع أو تطبيقات الويب الأخرى. ويحدث الهجوم من هذا النوع بمجرد أن يتقدم المتسلل للوصول المقصود إلى متصفح الويب الخاص بالمستخدم المصادق وقد يقوم باختطاف الجلسة وسرقة ملفات تعريف الارتباط وإعادة التوجيه الضارة وانتشار البرامج الضارة. كوسيلة لمنع مثل هذه الهجمات ، من الضروري تنفيذ تدابير أمنية تمنع بالتأكيد تدخل الطرف الثالث .في هذه الورقة ، نقدم آلية تحليل ودفاع ضد هجمات(XSS) نقترح فيها نموذجًا يسلط الضوء على نقاط الضعف الرئيسية التي تمكن هذه الهجمات ، ويوفر منظورًا مشتركًا لدراسة الدفاعات المتاحة. يتعلق البحث بمواقع الويب التي تم إنشاؤها باستخدام

أدوات مفتوحة المصدر تواجه هجمات تسمح للمهاجم بسرقة كلمات مرور المستخدم أو التحكم في جلسة المستخدم أو تشغيل تعليمات برمجية ضارة أو استخدامها كجزء من عملية احتيال (تصيد احتيالي). ثم قدم البحث حلاً لتخزين كلمات المرور والبيانات الحساسة باستخدام نموذج أمان جديد (NSM) يتكون من مفتاح استعلام سري (SQK) يمثل عشوائيًا سلسلة علنية مخزنة في قاعدة البيانات تتكون من تمثيل سداسي عشري يبلغ 8 بايت لكل منها إنشاء كلمة مرور المستخدم. يؤدي ذلك إلى تجزئة كلمة المرور باستخدام مفتاح الاستعلام السري (SQK) بحيث يمكن تخزينها بأمان في قاعدة بيانات. ثم نقوم بتشفير إخراج هذا البيان (سلسلة سداسية عشرية 64 بايت) تمثل تجزئة باستخدام خوارزمية SHA-256 لكلمة المرور ذات 32 بايت. ينتج عن الإخراج بعد التشفير باستخدام SHA-256 كلمة مرور قوية لا يمكن استردادها من التجزئة. يتم دمج تقنية تشفير التجزئة هذه مع عبارات الإجراءات المخزنة التي يتم إنشاؤها بواسطة عبارات الاستعلام المسبق لإنتاج نموذج أمان قوي (NSM) لغرض تشفير وفك تشفير كلمات المرور أثناء عمليات التسجيل وتسجيل الدخول والمصادقة ، ينشر هذا النموذج أمانًا جديدًا وتقنية فعالة للغاية وتمنع تمامًا هجمات البرمجة النصية للمواقع(XSS) .

ABSTRACT

To provide security for web applications in recent years is considered an obsession for many companies and users. The Cross site scripting (XSS) attacks always occupy the top most priority over other site or Web application threats. Once an intruder advances intended access of the authenticate user’s web-browser and may perform session hijacking, cookie-stealing, malicious redirection and malware spreading. As prevention against such attacks, it is essential to implement security measures that certainly block the third party intrusion. In this paper, we provide an analysis and defense mechanism against XSS attacks. We propose a model that highlights the key weaknesses enabling these attacks, and that provides a common perspective for studying the available defenses. The research concerned with websites created by using open source tools that face an attacks which allow attacker to steal user

passwords, take control of a user's session, run malicious code, or be used as part of a phishing scam. Then the research present a solution of storing passwords and sensitive data Using new security model (NSM) that is composed from a secret query key (SQK) randomly represents a publicly string stored in the database that consists of hexadecimal representation of an 8 byte for each user password generated. This hashes the password with the secret query key(SQK) so that it can be stored securely in a database. Then we encrypt the output of this statement (64 byte hexadecimal string) representing the 32 byte sha256 hash of the password. The output after encryption using SHA-256 cryptographic function generates a strong password that cannot be recovered from the hash. This hash encryption technique is merged with stored procedure statements which are generated by recomputed query statements to produce a strong security model (NSM) for the purpose of encrypting and decrypting passwords during, registration ,login, and authentication processes, This model deploying a novel security technique that is very effective and completely prevents cross site scripting (XSS) attacks.

Keywords

Web Applications, XSS Injection attacks, Secret query key (SQK) , enhance encryption technique(EET), Runtime attack prevention.

1. INTRODUCTION

Web security is a major concern. In every web application the website is vulnerable to many attacks. The majority of the web applications and web sites are created using tools such as PHP and MySQL as an open source tools for developing them[1]. Open Web Application Security Project (OWSAP) has listed the SQL injections among the top 10 vulnerabilities threats and Cyber Attacks for the year 2021 [2], also reported in the Latest published study by Check Point company regarding "SQL Injection

Trends", here the researches came to know through Statistics SQL Injection logs ratio from one monitored network in the past 60 days (over 8000 events),This forces the attacked server to host advertisements, according to monitoring reports came from reddit.com company [3].

Also Acunetix yearly testing report for the Web Application Vulnerability portrays the state of the security of web applications and network perimeters. This year's report contains the results and analysis of vulnerabilities detected over the 12-month period between March 2019 and February 2020, based on data from 5,000 scan targets [4]. This analysis mainly applies to high and medium severity vulnerabilities found in web applications, as well as perimeter network vulnerability data.

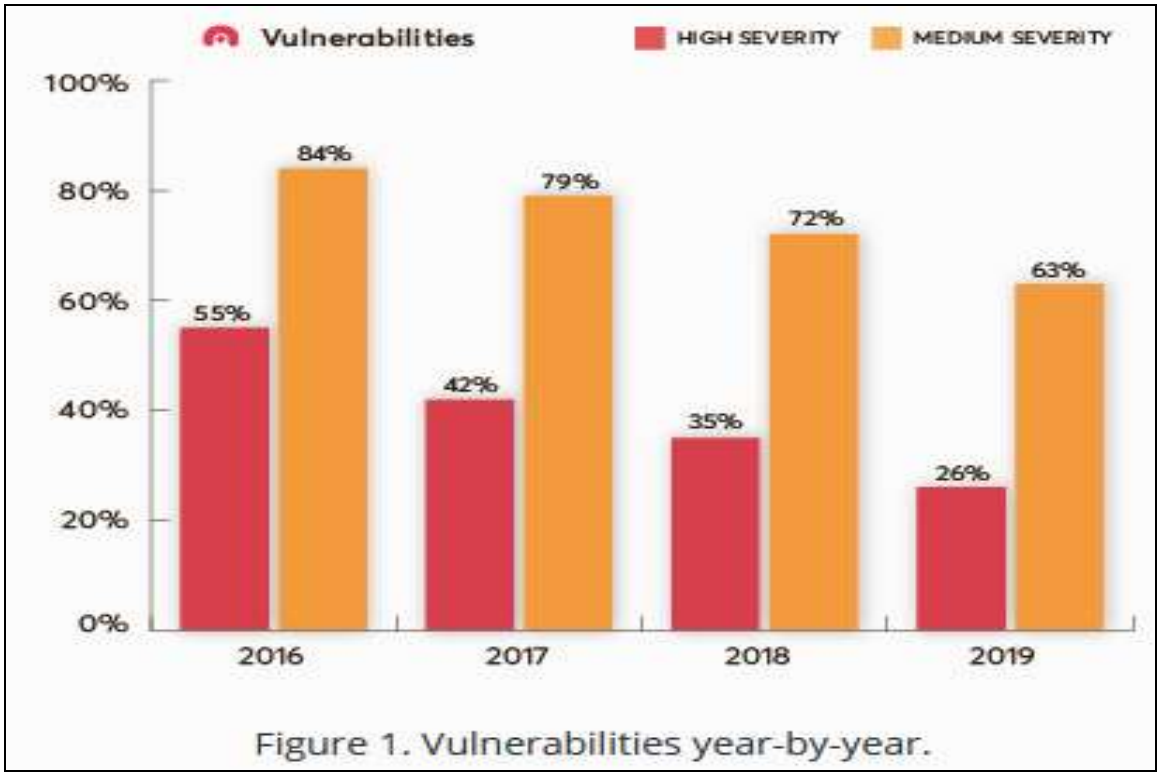

Figure 1. Vulnerabilities year-by-year.

Figure 1 vulnerabilities yearly impacts (2019).

As shown in figure 1 For the year 2019 report contains the results and analysis of vulnerabilities detected over the previous 12 months, across 10,000 scan targets[3.1]. Show that Cross-site Scripting (XSS) vulnerabilities, vulnerable JavaScript libraries, and WordPress related issues were found to each claim a significant 30% of the sampled targets.

The goal of the present paper is to find out the vulnerabilities related to XSS and other SQL injection attacks, provide the prevention techniques for these attacks. The aim is to create a sample login website to perform the various kinds SQL injection attacks to get authentication, perform the variety of operations related to select ,insert, update, and provides the prevention method for the particular type of attacks.

It is expensive for a Web site to require authentication, so it is usually only required when the site stores valuable private information. Corporate intranet sites can contain confidential data such as project plans and customer lists. E-commerce sites often store users' email addresses and credit card numbers. Bypassing or evading authentication in order to steal this data is clearly high on a hacker's priority list, and today's hackers have a large library of authentication evasion techniques at their disposal [1]. Simple web application Architecture is shown in figure 2 to make better understand on how Web application technologies interact.

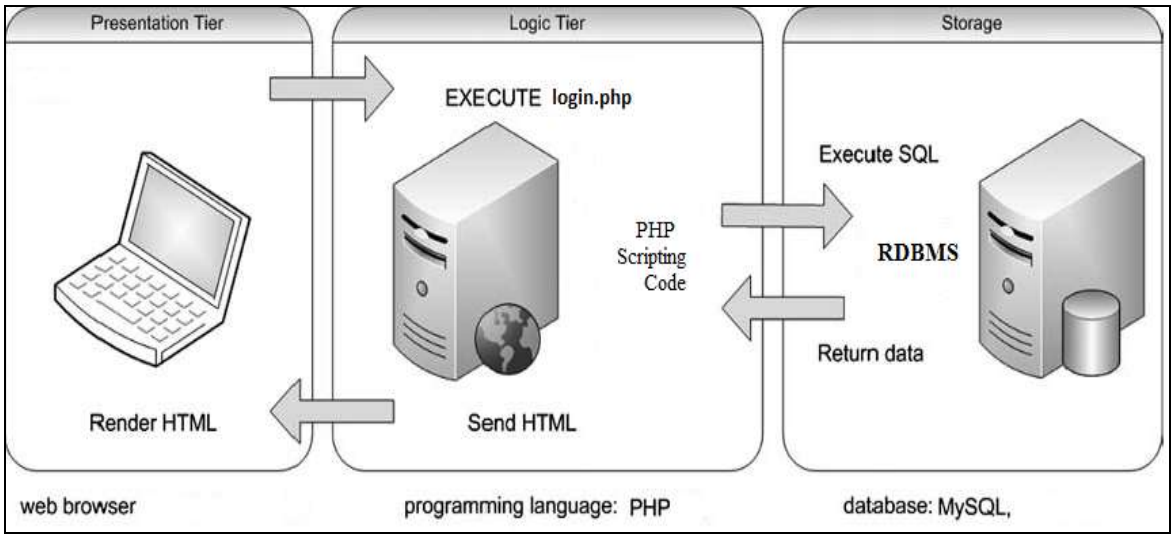

Figure 2 Simple web-Application Architecture [1].

2-XSS ATTACKS BACKGROUND

The SQL injection is the code injection technique, in which malicious SQL statements are inserted into an entry field for execution and used to attack the database and performs different types of the database interaction, operations and functions without sanitizing the inputs in the entry field[5] .

The SQL injection generally fools the database as a regular query by the user and gets access to the system easily. We can explain various types of SQL injection attacks, to give an idea about their impact and the serious results due to them .

Web application vulnerabilities and threats statistics for 2019 report generated by the Open Web Application Security Project. was rated the number of attacks on the OWASP top ten.[2] inform that the most commonly encountered web application vulnerabilities in 2019 involved Security Misconfiguration. One out of every five tested applications contained vulnerabilities allowing the hackers to attack a user session, such as sensitive cookies without the Http Only and Secure flags. Attackers can use such flaws to perform Cross-Site Scripting (XSS) in order to capture the user's session identifier and impersonate the user in the application.

Broken Authentication was found in 45 percent of web applications. Almost a third of such vulnerabilities consist of failure to properly restrict the number of authentication attempts. An attacker can exploit this to bruteforce credentials and access the web application. For instance, one of the applications could be accessed with administrator rights after only 100 attempts.

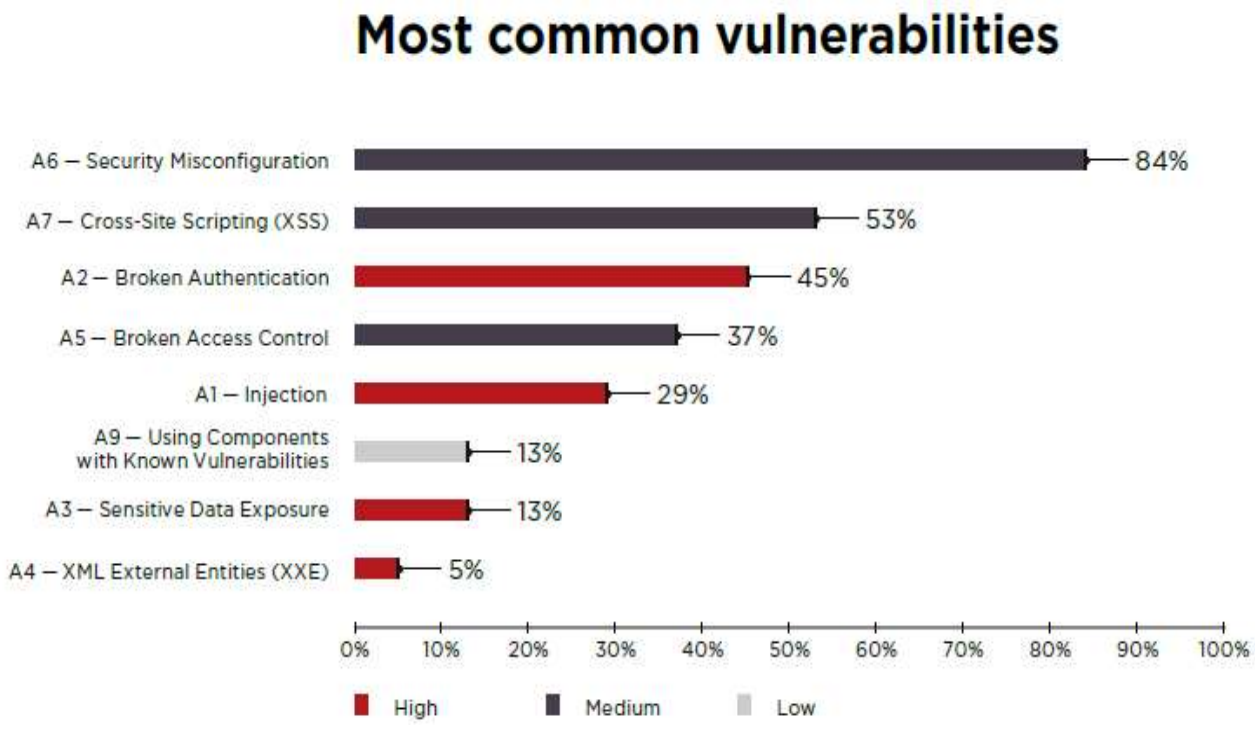

Figure 3 . show most common OWASP Top 10 vulnerabilities (percentage of web applications) year 2019/2020

3- RELAITED WORKS

Web security researchers around the world have interest in developing many techniques. The following reviewed can cover a part of the SQL Injection vulnerabilities types. and some important work on SQL injection attacks and their related prevention methods for the several last years is presented as follows.

A. Pooja Saini1, Sarita (2015), scheme In this paper, a hash function based authentication scheme including data validation is proposed to protect web applications against the SQL Injection attacks. The scheme proposed a Hash function algorithm and strong validation rule to provide Database Security from SQL Injection attack[7].
B. Thomas et al.'s Scheme (2009), This scheme proposed an automated prepared statement generation algorithm to remove SQL injection vulnerabilities in web applications [8]. The scheme implemented in this research work using four open source projects namely: (i) Net-trust, (ii) Itrust, (iii) WebGoat,and (iv) Roller. On the basis of the empirical results, their prepared statement codes were able to successfully replace94% of the SQL injection vulnerabilities in four open source projects.
C. I. Lee , S. Jeong, S. Yeoc, J. Moond Scheme (2011).
In this scheme the Authors proposed a technique to detect SQL injection attacks in web applications based on static and dynamic analysis [9]. This method removes the attribute values of SQL queries at runtime (dynamic method) and compares these values with the SQL queries analyzed in advance (static method) to detect the SQL injection attack in web applications. When run the application each dynamic generated query is compared or performed XOR operation with fixed query, if it results zero, then that particular query allowed to the database and if it results to non-zero then that query reported as abnormal query stop sending to the database of information system of an organization.

D. YashTiwari ,MallikaTiwari Scheme (2011).

This scheme proposed a technique to prevent various kinds of SQL injections,

by provide access to the database using an account with as less privileges

that are necessary and assurance of data validation with proper use of stored

procedures and parameterized queries using ORM framework can secure

the data from the intruders. Also design queries and using advance detections

gives better security and can handle the major threat of SQL injections[10].

E. William G.J.Halfond et al.'s Scheme (2005) This approach works by combining static analysis and runtime monitoring. In its static part, the technique uses program analysis to automatically build a model of the legitimate queries that could be generated by the application. In its dynamic part, technique monitors the dynamically generated queries at runtime and checks them for compliance with the statically generated model. Queries that violate the model represent potential SQLIAs and are thus prevented from executing on the database [11].

- **Analysis and critique of the previous works.**

The major differences between these methods and the research paper which developing the prevention techniques are:

i- the paper produced two protecting techniques cryptography hash Secret query key (SQK) and Parameterized query statement combined together to increase the level of security and it provides a successful technique to prevent two types of SQL injection attacks (bilateral way) .However those previous researches and papers produced a single protection techniques (unilateral way) or methods used to prevent the attacker from stealing passwords from the database.

ii- If intruder gets access to the system that uses brute force attack method or compound attack methods those techniques will not provide the required

iii-This research produces good security model for detecting and preventing cross site scripting attacks (XSS) as well as minimizes brute force vulnerabilities Security measure.

4. METHODOLOGY

As mentioned in the section (SQL injection attacks Background) the first issue comes to the developer mind for each open source application or web site is how to build a secure login Authentication mechanism to Protect Web Applications against SQL Injection Attacks , the web users especially ecommerce sites are much worried about security levels on the required application. This paper produces a new model for securing the authentication techniques to prevent blind SQL injection attacks from type cross site scripting attacks (XSS) attack ,one of the most common injection attacks types, then the paper concentrates on how to find a final treatment for a cross site scripting (XSS) attack, which is define as a technique for defeating a cipher or authentication mechanism by trying to determine its decryption key or passphrase by trying hundreds or sometimes millions of likely possibilities, such as words in a cross site scripting attacks (XSS) [6]. To make the problem clear we give two cases for Authentication problems, and then explain how the proposed technique will manage and solve this problem in the coming sections.

Case1:

Case one shows the first scenario and discusses how to register a new user to the application using insert query to add readable string consist of (username ,password ,E-mail) to create one record in the database.

The PHP SQL statement used for inserting the new values to the users table is:

$query = " INSERT INTO users (username,password,email) VALUES (:username,:password,:email)";

Given Table 1 shows the sample records for table "users" inserted on the database "dbinject" without Encryption.

Table 1 users table.

ID	USERNAME	PASSWORD(encrypted use sha256)	E-MAIL
8	yellow	Y123	yellow@hotmail.com

12	hybrid	hy@hy	hy@test.com
14	سامى	Sami123	sami@mymail.com
18	superuser	Longpasswordisverygood	super@hotmil.com

Note: username can accept English and Arabic letters.

Figure 4 shows a sample of the registration problems that the user may face while registering his username ,e-mail ,password , the response of the application is tested when one of the three data field is left blank .In this case without "username", immediately the system stop sand gives an error message "please enter a username" ,the system also gives the same response when password or e-mail is missing .

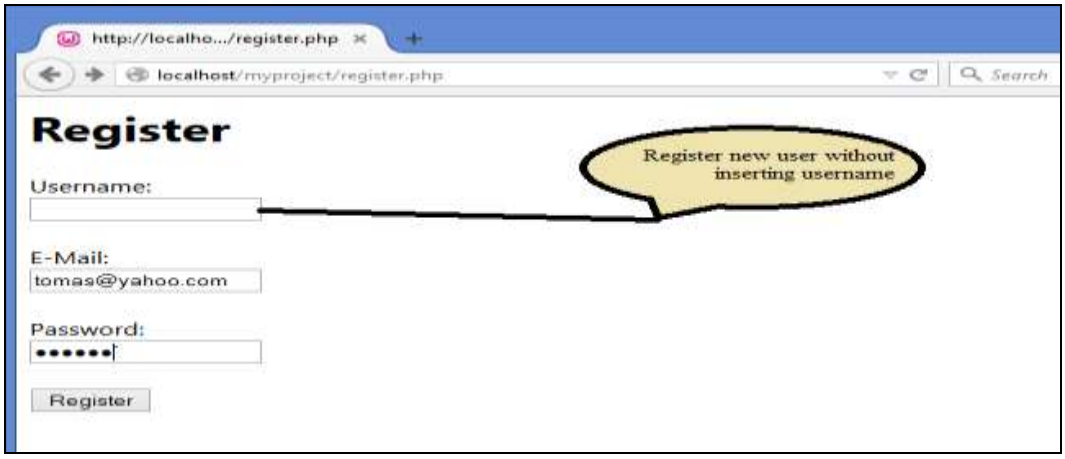

Figure 4 Registration problem.

One of the problems that faces the user is how to insert the proper data during the registration process , because it's considered one of the Threats that may face the Application, if the user inserting his data improperly . Figure 5 shows an example the application response with an error message.

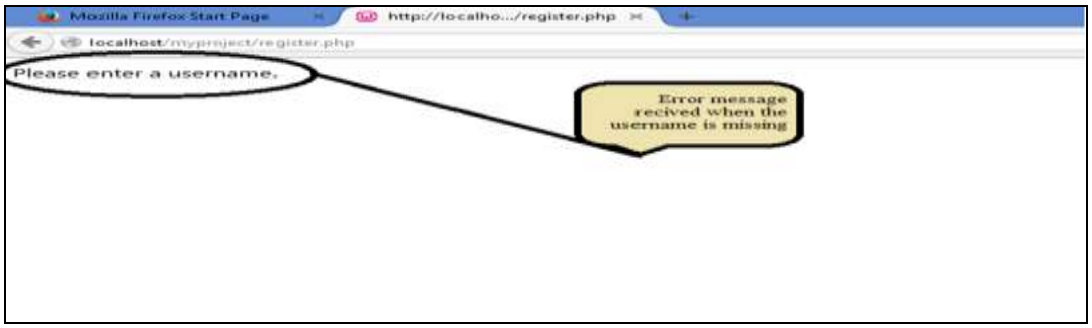

Figure 5 Registration error message.

Case- 2

The second scenario explains the Login authentication problem to the application which represents the most critical issue to all the web site developers and how to protect users of data against SQL injection attacks, for example suppose the following statement has been submitted to the login form then the PHP code is run to execute query in order to retrieve the user's information from the database using their username as explained by :

```
$query = "  SELECT     id, username, password, email
          FROM       users
          WHERE      username = :username  ";
```

In case-2 it is clear that using SQLI to execute SELECT statement to fetch user information from data base using the previous command make the application vulnerable to many kinds of SQLI attacks specially cross site scripting (XSS) attacks which is most preferred by the hackers ,as shown in case-1 and case-2 in the above scenarios , SQL injections is completely breakable when the attacker is using cross site scripting (XSS) attacks .Hence the study and tries to find final solution to prevent it, cross site scripting attacks some references also called them Rainbow table attacks ,this kind of attack used to recomputed table for reversing cryptographic hash functions, usually for cracking password hashes. Tables are usually used in recovering a plaintext password up to a certain length consisting of a limited set of characters.

4.1 ARCHITECTURE OF PROPOSED TECHNIQUE

As mentioned in the previous section we come to know how the problem exists ,when the hackers are selecting the cross site scripting (XSS) or pre-computed data table this will attack the data stored on the MySQL database server. In this section we explain how hybrid technique will work to stop the cross site scripting (XSS) attacks and keep the web site data base completely secure, in two phases plan.

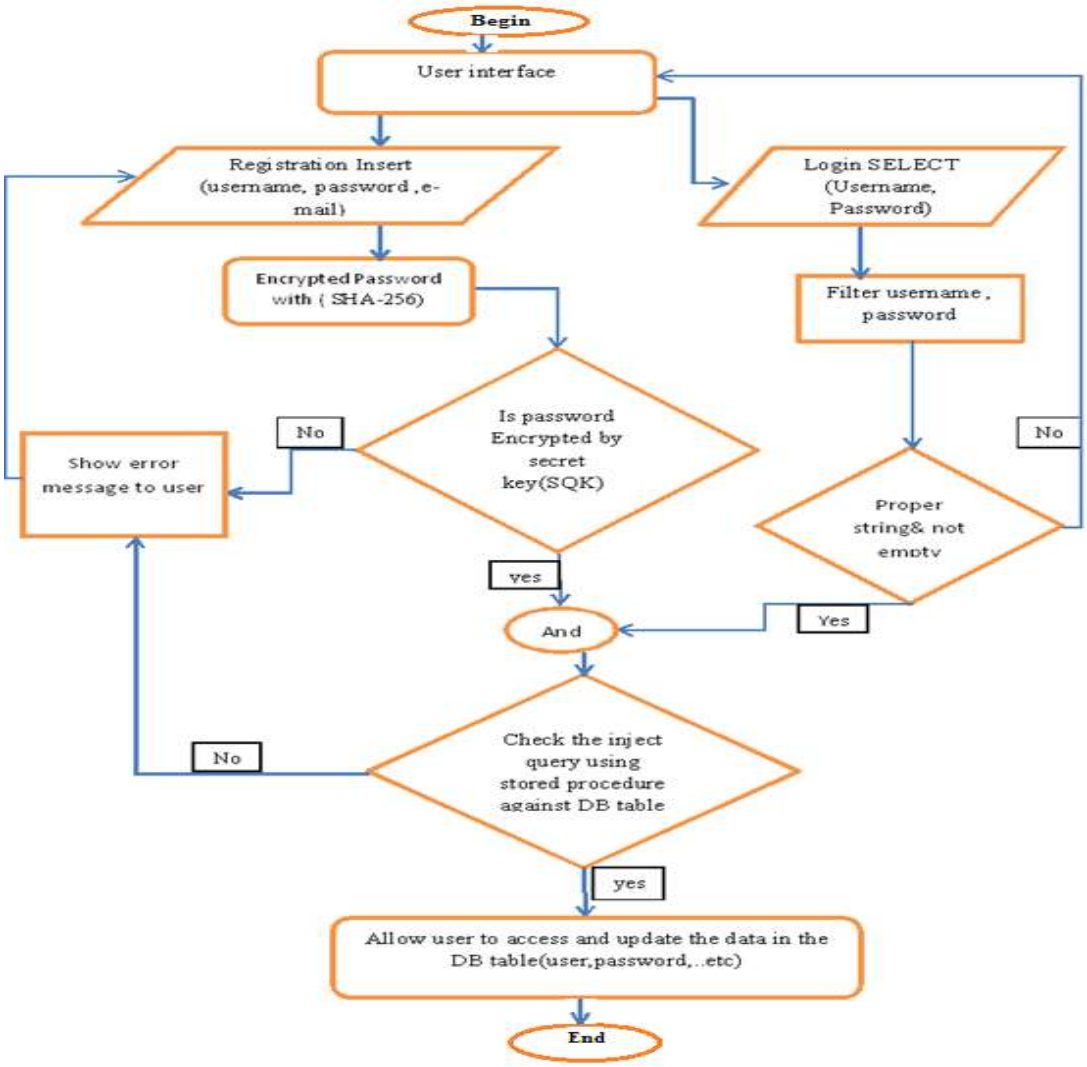

Figure 6 Architecture of proposed Hybrid Technique (security model)

Phse1: Build store procedure statement to inject the query

In this phase we take care of the database design and create data table in the database with some constrains. Features to control the query statements (select, insert), some of this constrains are :

1- Specify a common file to Organize and control the communication sessions.
2- Use UTF-8 is a character encoding scheme[12] that allows you to conveniently store a wide variety of special characters, like ¢ or €, including Arabic letter in a database.
3- Use PDO library in code designed to provide a flexible interface between PHP and database servers[13]

4- Write SQL query to see whether the username entered by the user is already in use or not . A SELECT query is used to retrieve data from the database use special token.
5- Use prepared statement to pre-compute the injected data before sent them to the server. This contains the definitions for any special tokens that we place in our
SQL query. In this case, we are defining a value for the token (username). It is possible to insert $_POST['username'] directly into your $query string; . However in doing so is very insecure and opens your code up to SQL injection exploits. Using tokens prevents this.

Phase2: Using hash function supported by secret query key (SQK)

The most dangerous part in the cross site scripting (XSS) attacks is when the attacker writing a successful SQL injection software for defeating a cipher or authentication mechanism by trying to determine its decryption key or passphrase by trying hundreds or sometimes millions of likely possibilities s, such as words in a cross site scripting (XSS).

In phase two an encryption technique is developed for preventing this kind of attack as follow

1- Design a database table with some security feature by automatically encrypting the password use SHA-256 hash function algorithm in the PHP scripting code.
2- A secret query key (SQK) is randomly generated here to protect against cross site scripting (XSS) or brute force attacks and rainbow table attacks[14] . The following statement generates a hexadecimal representation of an 8 byte secret query key(SQK). Representing this in hexadecimal provides no additional security, but makes it easier for humans to read.
$SQK = dechex(mt_rand(0, 2147483647)) . dechex(mt_rand(0, 2147483647));
This hashes the password with the secret query key(SQK) so that it can be stored securely in the database.
3- The output of this next statement is a 64 byte hexadecimal string representing the 32 byte sha256 hash of the password. The original password cannot be recovered from the hash.

```
$password = hash('sha256', $_POST['password'] . $SQK);
```

4- Next we hash the hash value 65536 more times. The purpose of this is to
protect against brute force attacks. Now an attacker must compute the hash 65537 times for each guess they make against a password, whereas if the password were hashed only once the attacker would have been able to make 65537 different guesses in the same amount of time instead of only one.

```
for($round = 0; $round < 65536; $round++)
    {
      $password = hash('sha256', $password . $SQK);
    }
```

5- Then we prepare our tokens for insertion into the SQL query. We do not store the original password; only the hashed version of it. We do store the secret query key(SQK) (in its plaintext form; this is not a security risk).
Table 2 below explains how the proposed method covering all the levels of application starting from the storage level where the database tables and data records are stored.

Table.2 Users Secret Query Key(SQK), password encryption with SHA256

ID	User	passwrd(string)	password ecrypted using SHA-2(256)	SQK	E-mail
14	سامي	Sami123	136b23f61bc7a91b86c19063d8178734d457d54d7cc4f87e9ff2ab954b6c96396	1dbaff181726357447	sami@mymail.com
18	superuser	Long passwordisverygood	96d952ffffe65ffe15741ff7203e53e8b3c8feb6b07ff04163bb43301d91261ff0389	5529ede41f1b6def	super@hotmail.com

6- Finally Execute the query to create the user, with a strong , secure and encrypted password.

5. METHOD OF GENERATING HYBRID TECHNIQUE

After finishing phase 1 and phase 2 ,the application became ready for testing all the mentioned features in both phases.

The last step in the proposed technique is to combine the two phases in order to verify all the security items and how they interact together to prove secure authentication during the run time of the application .

The combined testing results for the two major security procedures which are studied by the proposed technique in this paper is producing Hybrid protection technique.

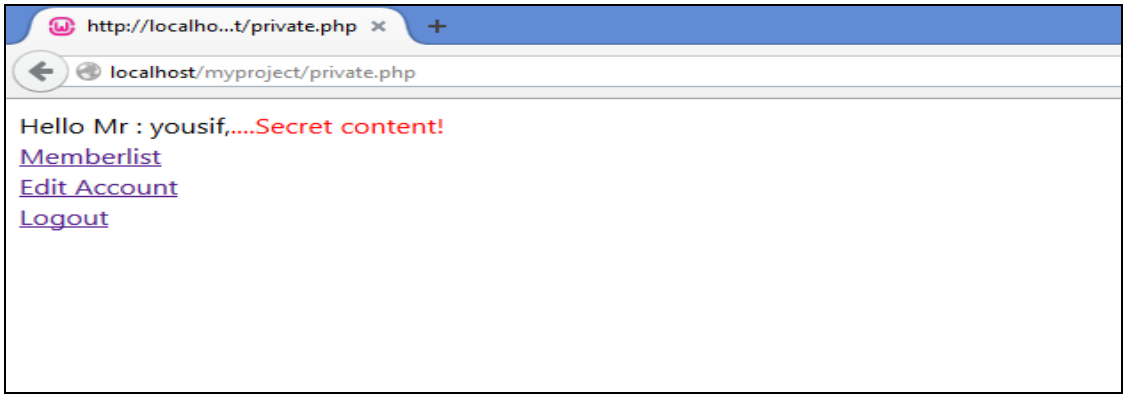

Figure 7 the proposed hybrid techniques login authentication

The combination of the two phases in one PHP code is also increases the security level of the data in database table.

On the other hand the proposed technique also prevents to some extent another type of SQL injection that known as Cross-site scripting or XSS as shown on figure 7 .According to WhiteHat Security Statistics Report (2017) [15], Developers are prioritizing easy fixes over the tougher, more serious vulnerabilities. This report is rates by developers – Application Misconfiguration (a rate of 74%), Insecure Digest (66%), and Unpatched Library (62%) – are all easy fixes. While the toughest, most complex to esolve – XSS (38%) and SQLi (32%) – are not being addressed adequately, so preventing such kind of attack becomes necessary, because without it users can view your members-only content without logging in which is very dangerous. For this reason the paper also considers the overlapping between XSS attack and dictionary attack to resolve both of them simultaneously to avoid any overlapping threats[16].

This hybrid technique prevention is generated by writing an intelligent PHP scripting codes supported by hash encryption function to keep the user session safe and prevent redirect or go back to the previous page, we can

say by using this technique the user kept save from XSS attack automatically according to application behavior .

6-RESULTS AND DISCUSSION

The performance of proposed technique has been evaluated on a table having different number of user records[17]. We computed number of total tries; of login to the targeted application then we make a comparison between the successful trials and the fail trails as Table (3) given below:

Table 3 – Performance analysis of proposed technique**.**

IE	Mozilla	Opera	NSM model(XSS)
2	5	20	50
10	8	47	100
20	10	12	200
30	15	11	300
27	0	30	400

The table number 3 show the test results while tried three type pf internet browsers with and without using the new security model (NSM) to display the prevention activities.

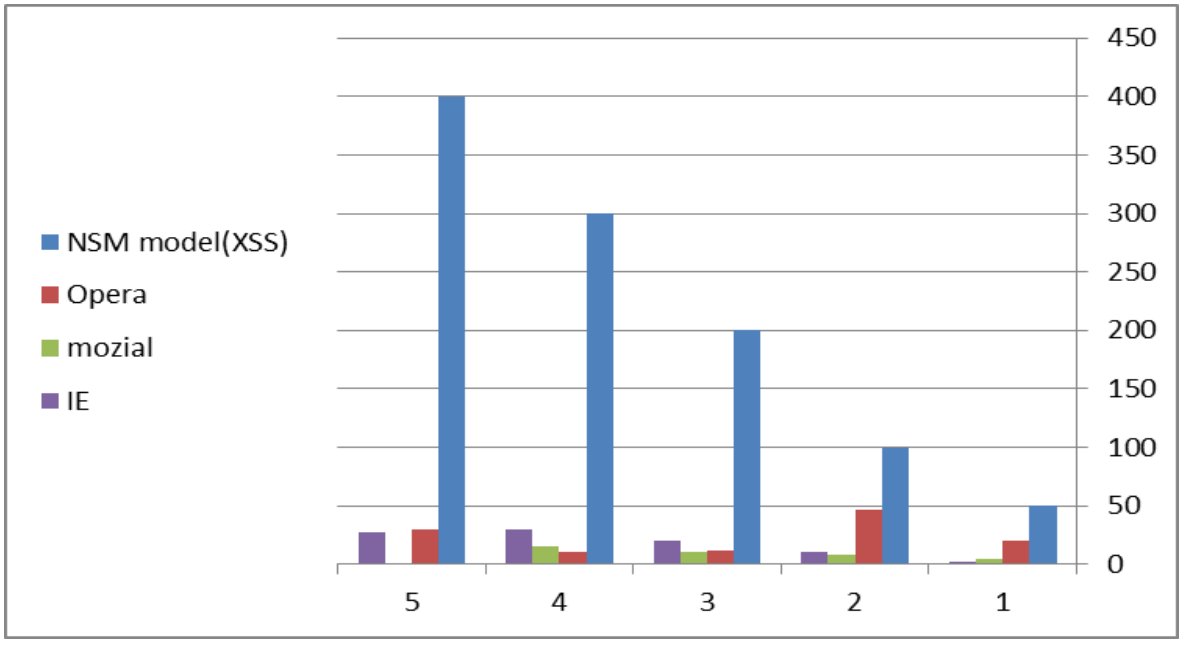

Figure 8 comparisone rates for four browsers aginst the (NMS) modle.

The figure number 8 show the graphical view of the test results for the three type pf internet browsers with and without the new security model

(NSM) , and it is clearly show that the cross site scribting attak new security model is affected in preventing unautherized login for the tree browsers (IE,Mozila ,Opera) with the sueccesful persent 78.41 %

7. CONCLUSION

new security model (NSM)provides a new method for protecting passwords and prevents attacker from stealing it . for every individual account they're attempting to crack. The proposed security model (NSM)not only helps against XSS attack, but also preventing intruder from type dictionary attack, and also prevents the application from brute force attack , also using intelligent PHP scripting codes kept user session safe and prevent redirection, so the user will be save from XSS attacks automatically with a high present reaching 78.41 % .

8. FUTURE SCOPE

In the present time, web applications must provide full security and assurance to the users. During the review of previous research , we found that in certain cases, these approaches were not fully effective. Hence ,these approaches are not useful for detecting and preventing multiple SQL injection attacks or compound attacks types .

The proposed technique can only protect authentication mechanism of web applications from type XSS attacks . So, in future, we will try to improve the technique by making it more secure and efficient for other types of SQL injection attacks . by adopting more protection methods and security techniques .Then, this technique will be able to prevent SQL Injection Attacks completely.

9. ACKNOWLEDGMENT

This joint research achieved by Dr, Khalid Ahmed Ibrahim and Dr. Yousif AbdElmalik GasmElseed Mohamed also we would like to convoy sincerely thankful to Dr. Mohamed Osman Ali Hegazi Associate Professor in Faculty of Computer Science and Information Technology, Alzaiem Alazhari University, for his valuable advice. .

10. REFRENCES

[1] Justin Clarke."SQL Injection Attacks and Defense".Syngress Publishing , Burlington,pages(2-4,6-8),1th Edition 2009 .

[2] https://owasp.org/Top10/A03_2021-Injection/ .

[3] Adi Kaploun and Eliran Goshen.,(2015) . 'The Latest SQL Injection Trends', Check Point Threat Intelligence & Research Team posted 2015/05/07 http://blog.checkpoint.com/2015/05/07/latest-sql-injection-trends/.

[4] https://www.acunetix.com/white-papers/acunetix-web-application-vulnerability-report-2020

[5] Abhay K.Kolhe , Pratik Adhikari. " Injection, Detection, Prevention of SQL Injection Attacks ", International Conference on Collaborative Computing: Networking, Applications and Worksharing , 978-1-63190-043-3 ICST DOI 10.4108/icst.collaborate com.2014.257568 ,(2014).

[6] https://en.wikipedia.org/wiki/SQL_injection

[7] Pooja Saini1, Sarita" Authentication Scheme to Protect Web Applications against SQL Injection Attack Using Hash Functions ", International Journal of Advanced Research in Computer and Communication Engineering ISSN 2278-1021 Vol. 4, Issue 6, June 2015

[8] S. Thomas, L. Williams, and T. Xie, "On automated Prepared statement generation to remove SQL injection vulnerabilities". Information and Software Technology, Elsevier .51, 589–598, 2009.

[9] I. Lee , S. Jeong, S. Yeoc, J. Moond, "A novel method forSQL injection attack detection based on removing SQL queryattribute", Journal of Mathematical and Computer Modeling,Elsevier 2011.

[10] YashTiwari ,MallikaTiwari ,"A Study of SQL of Injections Techniques and their Prevention Methods " ,International Journal of Computer Applications (0975 – 8887) Volume 114 – No. 17, March 2015.

[11] William G.J.Halfond and Alessandro Orso "AMNESIA:Analysis and Monitoring for Neutralizing SQL-Injection Attacks",Nov 7, 2005.

[12] https://en.wikipedia.org/wiki/UTF-8.

[13] Vikram Vaswani ,(2007).'PHP programming solutions ', by The McGraw-Hill Companies, ISBN: 0-07-159659-3.

[14] Matt Curtin.,(2005) .'Brute Force Cracking the data Encryption Standard', Copernicus Books Springer :ISBN 0-387-20109-2.

[15] https://info.whitehatsec.com/Content-2017-Stats-Report-LP.html?utm_source=website&utm_medium=Website-Content-2017-StatsReport.

[16] Jeff Atwood “Dictionary Attacks”, on line report , 07 Jan 2009 https://en.wikipedia.org/wiki/Dictionary_attack#cite_ref-1

[17] M.F.G.Matthewman, M.Ed., F.C.P,"Examination Results, Processing, Analysis and Presentation". RoutledgeFalmer ,1th Edition2000.