International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 05 | May-2016 www.irjet.net p-ISSN: 2395-0072

Novel Approach To Detect and Prevent Web Attacks

Mrunali P. Pathak [1], Nida Kausar Khan [2], Tejashree C. Tantak [3]

[1]*Mrunali P.Pathak Pathardi Phata,Nashik-422009*
[2]*Nida Kausar Khan Wadala Pathardi Road-422009*
[3]*Tejashree C.Tantak Canada Corner,Nashik-422002*
[4] *Professor. R.S.Jagale Dept. of Computer Engineering, G.E.S.R.H.SAPAT College of Engineering, Maharastra, India*

Abstract - *Most of the cyber security techniques present today have many critical faults. This provides to be an open way for hackers and criminals. Using widespread techniques, hackers and criminals try to gain entry into one's system using these backdoors. However, gaining entry and retrieving information can be a tedious task for the hackers. Hence they use techniques such as SQL injection and Cross-Site Scripting (XSS) to obtain sensitive data such as password, account no etc. This paper analyses the source code of security patches of widely used web applications written in weak and strong typed languages. Results show that only a small subset of software fault types, affecting a restricted collection of statements, is related to security. To understand how these vulnerabilities are really exploited by hackers, this paper also presents an analysis of the source code of the scripts used to attack them. The outcomes of this study can be used to train software developers and code inspectors in the detection of such faults and are also the foundation for the research of realistic vulnerability and attack injectors that can be used to assess security mechanisms, such as intrusion detection systems, vulnerability scanners, and static code analyzers.*

***Key Words*: Intrusion detection, XSS, SQL injection, hackers, criminals, vulnerability, click-jacking, tab-nabbing.**

1.INTRODUCTION

Before discussing the concept of web-attacks, it is important to know how the web has evolved over time. The first generation web-based applications were very limited in their ability to cater to the needs of the user. To bridge this gap, the web applications evolved to give services like searching, posting and uploading. CGI, or Common Gateway Interface protocol provided a path to the user to interact with the web pages by submitting data into forms.

CGI scripts would process the information at the back end and then submit it back to the end-user. This provided the very first attack vector for the web applications. Along-with CGI scripts, various frameworks like AJAX, Ruby on Rails, PHP, ASP.NET, and J2EE were developed to provide flexibility and manage workflow in web services.

Web Attack is simple an attack on a computer through the medium of a web based services like a website. The purposes of a web attack are varied and can be used for different reasons. So what is a web attack made of? The basic web attack consists of a modified request that is designed to take disadvantage of the user's trust.

Web attack often compromises of various stages but in general, the basic 5 stages of a web attack are as follows:

1. **Entry Point:** The attacker visits a legitimate website
2. and uploads a malware on the website. When the user visits this website, the malware uploaded on the website gets downloaded on the user's system without his/her knowledge.
3. **Distribution:** The malware will redirect the user to an exploit server depending on the user's OS.
4. **Exploitation:** Supported exploit packages will try to execute the vulnerabilities in the OS, browser, plugins etc.
5. **Infection:** The malware downloaded on the user's system will download a malicious payload on the system that will steal sensitive data from the user's system or session.
6. **Execution:** The malware will execute at the backend and trick the user into doing something without his knowledge.[6]

2. Literature survey

Jose Fonseca.*et.al.* The author have discussed the typical web faults are discussed considering various programming languages. In order to solve the problem of web security vulnerabilities, the authors have inspected the code patch to gather characteristics of code responsible for vulnerable behavior.[1]

S.Shalini.*et.al,* The authors have inspected the worms exploiting JavaScript XSS vulnerabilities. To counter this exploitation, the authors have suggested a client side solution that uses a step by step approach to protect cross site scripting, without degrading the browsing experience.[2]

Lwin Khin Shar *.et.al,* The authors have presented the current approaches to mitigate this problem mainly focus on effective detection of XSS vulnerabilities
in the programs or prevention of real time XSS attacks. To solve this problem, the authors have proposed method consists of two phases:
(1) XSSV detection and
(2) XSSV removal.
XSSV detection phase identifies potential XSSVs in the program source code using static analysis. XSSV removal phase first determines the context of each user input referenced in the identified potential XSSVs. It then secures the potential XSSVs by applying the appropriate escaping methods using an escaping library provided by ESAPI.[3]

William G.J. Halfond.*et.al,* This paper presents a new highly automated approach for protecting Web applications against SQL injection that has both conceptual and practical advantages over most existing techniques. The authors have stated that detecting.[4]

Kuldeep Kumar.*et.al,* In this paper, we proposed a method based on grammatical structure of an SQL statement and validation of the user input. This method uses combined static and dynamic analysis. The experiments show that the proposed method is effective and simple than other methods.[5]

Problem Definition:

Design and implement a system such that it detects and prevents a potential Web Attacks.

3. Architectural Design:

In our project we are developing the prevention system for web attacks. Web sites are used in todays era by every user for online shopping, bill pay, online recharge and for many more purpose where secure data is used like credit card number, password.etc which may be hacked. So, it is important to detect and prevent the web attack.

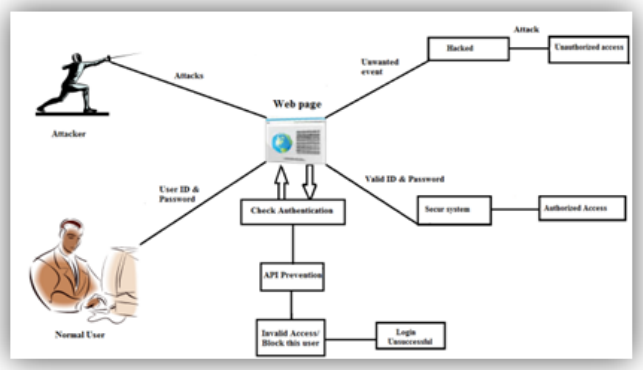


Fig -1: System Architecture

While studying the various web attacks and the components that were implemented in our project, we realised that the overview of every attack is nonetheless very similar in nature. Hence we have kept the general architecture of the project the same. It undergoes very minute changes during the implementation of module .The solutions on server side result in considerable degradation of web application and are often unreliable, whereas the client side solutions result in a poor web browsing experience, there is need of an efficient client side solution which does not degrade the performance. The proposed system is designed in order to provide effective security against the Cross Site Scripting attack, keeping the concept of usable security with optimized web browsing. This approach uses a three step process.

4. Types of Web Attacks:

With the development of the Web and the websites on the web, the attackers found out many new attacks to harm the confidentiality of websites. Some of these attacks are non-existent but most of them are still working. The most common web attacks are as follows:

4.1 Cross-Site Scripting Attacks (XSS Attacks):

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side script into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy. Cross-site scripting carried out on websites accounted for roughly 84 percent of all security vulnerabilities documented by Symantec as of 2007.

Their effect may range from a petty nuisance to a significant security risk, de-pending on the sensitivity of the data handled by the vulnerable site and the
nature of any security mitigation implemented by the site's owner. Cross-site scripting attacks use known vulnerabilities in web-based applications, their servers, or plug-in systems on which they rely.

Exploiting one of these, attackers fold malicious content into the content being delivered from the compromised site. When the resulting combined con-tent arrives at the client-side web browser, it has all been delivered from the trusted source, and thus operates under the permissions granted to that system. By finding ways of injecting malicious scripts into web pages, an attacker can gain elevated access-privileges to sensitive page content, session cookies, and a variety of other information maintained by the browser on behalf of the user. Cross-site scripting attacks are therefore a special case of code injection.

A malicious script can be embedded in the source code of the web page, the data that is submitted through forms available on the website or through any other means. This script can be designed for various purposes but are mainly used to steal data from the user or the server. The script is a specially crafted hyperlink, usually written in JavaScript. The JavaScript has the ability to influence the behavior of the browser.

The problems that can arise due to JavaScript are as follows:-

- Making changes to the local system, like changing the file system, copying data etc.
- Identifying keystrokes to steal confidential data like passwords
- Interact with the websites opened in the browser tabs without the user's knowledge.
  The Cross-site Scripting attack takes the advantage of the vulnerabilities present in the website by implementing dynamic elements to compromise the security of the website.

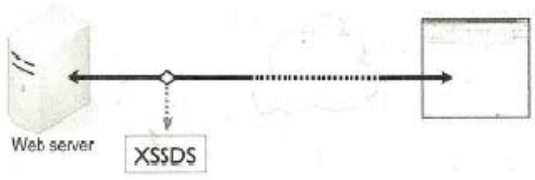


Fig-2: Overview of XSS Attack

Web browsers are designed in such a way that they do not let the website access documents loaded from other sites. Hence the attackers have to implement the XSS using different means. There are main three types of XSS attacks:

1. **Non-persistent Attacks:** This is the most common type of XSS attack. This attack targets the vulnerabilities present in the website when the data submitted by the client is processed at the server side to produce some results and the results are sent back to the client browser. The attacker tries to embed the malicious code in the results that are sent to the client browser. The attack is successful only when the malicious code is encoded into the Web page results, the results are sent to the client browser, where the code is not encoded using HTML special character coding and is not detected as inert text. The most common way of implementing an XSS attack is a link that involved badly-written URL.
2. **Persistent XSS Attack:** This attack works when the exploit is stored in the database at the server side. This exploit can be used to create malfunctioning webpages that will be displayed to other users later. The example of stored XSS attack is the forums and bulletins which allow the user to use HTML and XHTML to format their posts.
3. **Dom-based XSS Attack:** This is an unusual type of XSS attack wherein the logical errors in legitimate JavaScript and careless handling of client data results in XSS attack conditions.[1]

Cross-site scripting attack used the javascript code which actually creates an attack on the web site.

Example of Javascript code for attack:

1.<script>alert(0)</script>

2. <script> window.open ("http://evilsite.com/cookie_stealer.php?cookie=" + document.cookie, "_blank"); </script>

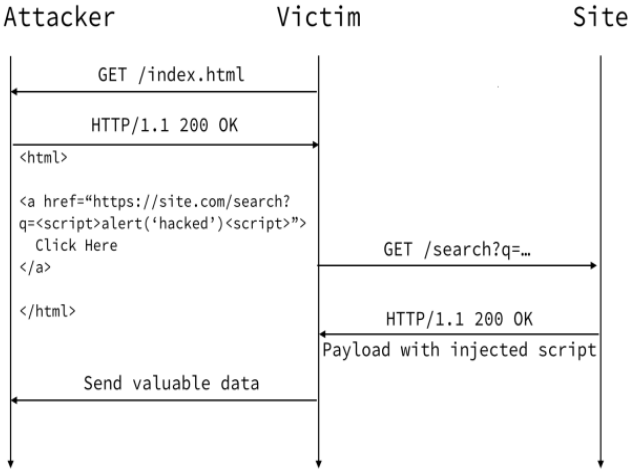


Fig-3: Non-Persistent XSS Attack Scenario

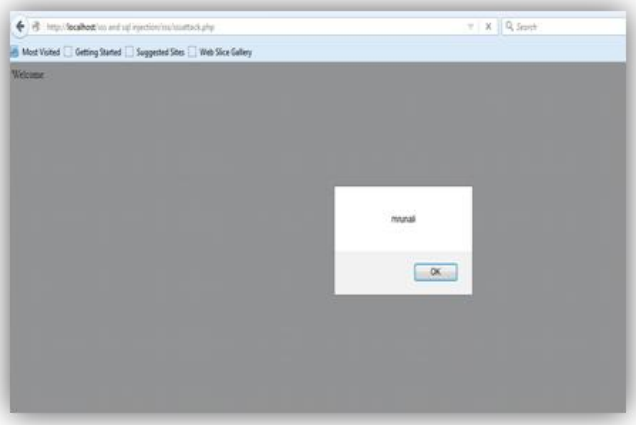


Fig-4*:* Working of XSS Attack

There are many challenges in defending the system against XSS attack which are listed as follows:

1. **Encoding Schemes**: If the URL is in an encoded form, it is not easily detected and hence the XSS attack can be easily implemented.
2. **Anti-comment design flaws:** The malicious code can be immediately filtered out automatically using the Auto-comment feature. This feature blocks the malicious code by commenting it.
3. **Browser versions:** Various browsers have various vulnerabilities. The modern day browser versions have an in-built capability of preventing the XSS attacks.
4. **Tags:** XSS code that is embedded in some tags can execute automatically if sent by e-mail.

4.2 SQL Injection:

SQL injection is a code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker).[1] SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

SQL injections allow the attackers to obtain unrestricted access to the databases running at the background of a web application. Due to this attack, the attacker can access financial as well as personal data of the user. SQL Injection attack takes place when the attacker changes the intended query by inserting new SQL keywords or operators in the query. There are various methods through which an SQL injection attack can take place:

- **Injection through cookies:** If a website uses cookies, the attacker can easily embed an attack in the cookies used by a website to simulate an attack.
- **Injection through user input:** The attacker can easily modify the user queries to insert SQL commands that fetch the data needed by the attacker.
- **Injection through server variables:** Server variables are collection of variables that contain HTTP headers, network headers and environmental variables. The attacker can directly place SQL attack query in the headers. The attack simulates when the query executes to log the server variable to the database.
- **Second-order injection:** This type of attack is quite different from the other attacks. This attack executes when the input i.e malicious query stored in the database is used.

We shall now discuss the various types of SQL injection techniques known along-with the intent of the attacker and the description of the attack.

Some SQL injection attacks are also done using magic code.

Example:

Username= admin

Password= ' or 1=1 #

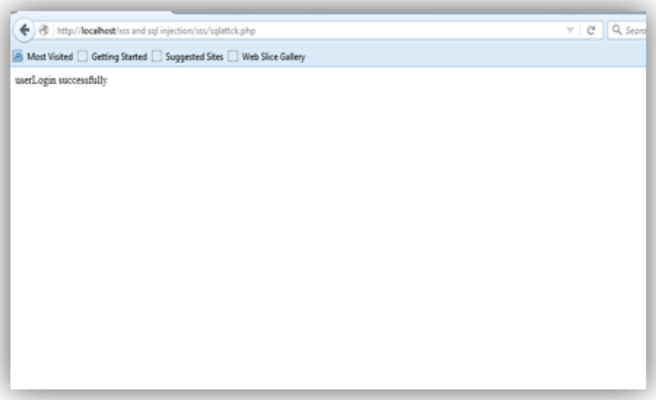


Fig-5: Working Of SQL Injection Attack

4.3 Click-jacking Attack

Clickjacking is a web framing attack that has recently received wide media coverage. Web framing attacks such as clickjacking use iframes to hijack a user's web session. In a clickjacking attack, a malicious page is constructed such that it tricks victims into clicking on an element of a different page that is only just or not at all visible.

It was introduced by Robert Hansen and Jeremy Grossman in September 2008. This attack constructs a

malicious web page to trick the user into performing unintended clicks that are advantageous for the attacker. Its propagate worms, steal confidential information passwords, cookies, send spam, delete personal mails, etc[1].

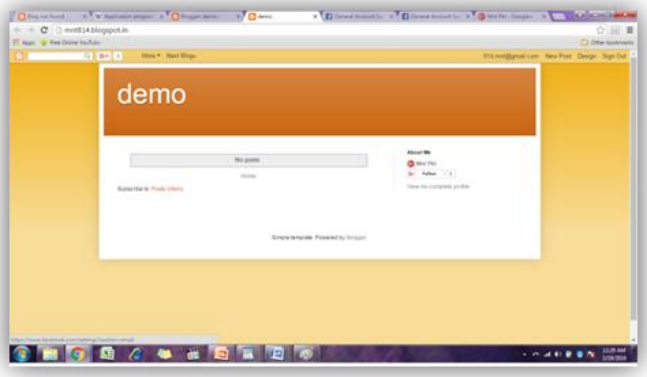
Clickjacking takes the form of embedded code or script that can execute without the user's knowledge, such as clicking on a button that appears to perform another function. Clickjacking also known as user interface redressing is one of Malicious Technique tricking users to click the button or image that will run hidden malicious script from another site. An attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the innocuous page [2]. Thus an attacker hijacks the click to another website. That's why it is known as Click-jacking (Click+Hijacking).

Click-jacking attack can be implemented in a more frequent manner due to the following reasons:

1. The program can run on virtually any Web site without the Web site owner's knowledge or ability to stop it.
2. Click-jacking can take the user to a mirror site while still making them believe they are on the Web site of the company and mine personal information, often which is freely given.
3. No browser, except the very few that are not based on graphics, is immune to a click-jacking attack.

Working of Click-jacking attack

A typical clickjacking attack uses two nested iframes to crop and position an element from a target website. The inner iframe contains the target page and must be large enough to display it in its entirety, such that the element on which the user will click is visible without scrolling. The outer iframe is much smaller and acts as a window onto the page loaded in the inner iframe. For a user interface redressing attack, the outer iframe should only be large enough to display the targeted element [13]. You think you are clicking on the website you see but no, you are really clicking on an invisible website you cannot see that's right under your mouse. Click-jacking affects many browsers and platforms.



.

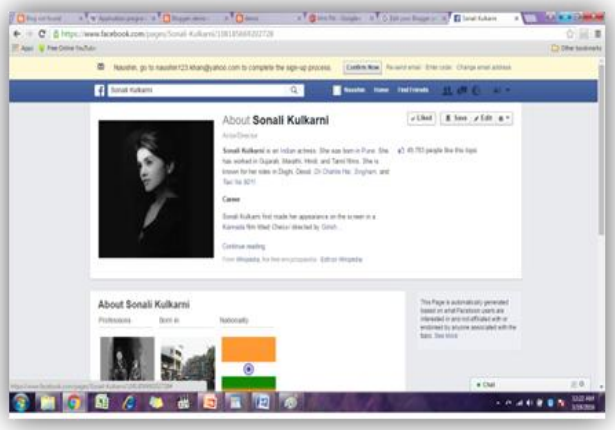


Fig-6: Working Of Click-jacking Attack

5. Prevention of Attacks

6.1 Prevention of Cross-site scripting (XSS):

XSS attack is a web attack that is uses script languages to retrieve information from the client. This script is injected into the webpage by the attacker and the script that is frequently used by the attackers these days is JavaScript. The effect of an XSS attack is largely dependent on the type of data which is handled by the website and can range from a mere prank to a huge security risk. The XSS attack typically uses the known vulnerabilities of the website, plug-ins and the servers to successfully in-corporate an attack. Popular websites such as Facebook, Twitter, MySpace, Youtube and Orkut are some of the known victims of XSS attacks. Here, we present you a bit of code that we have used to execute XSS attack using JavaScript:

```
$sql= insert into scriptrec (script) values ($name$);

mysql_query(sql);
```

For the prevention of XSS attack, we use PHP language which is implemented in the following manner:

```
<? php
   $name= htmlentities($_POST [name] );
?>
```

We implemented the prevention using a PHP function "htmlentities"which is a special function htmlentities()converts all characters which have HTML character entity equivalents are translated into these entities and returns the input string. one common method for carrying out an XSS attack involves injecting an HTML element with an src or onload attribute that launches the

attacking script. PHPs htmlentities() function (information is at http://php.net/htmlentities) will translate all characters with HTML entity equivalents as those entities, thus rendering them harmless. Its sibling htmlspecialchars() is more limited, and should not be used.

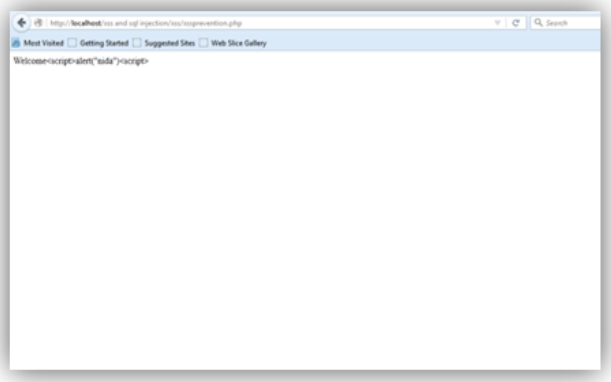


Fig-7: Working Of XSS Prevention

5.2 Prevention of SQL injections Attack

There are many ways to prevent SQL injections and some of the techniques have been summarised as follows:

- Checking input types
- Input encoding
- Positive pattern matching
- Identification of all input sources

The SQL Injection attack, popularly known as SQli is used to retrieve informa-tion from data driven applications and is implemented by inserting a malicious SQL query into an entry field. This entry field can be a field that is used to get the details of the user. By inserting this query into the field, that information stored at the back-end will be accessible to the attacker. The SQLi is executed in the following manner:

$sql = SELECT * FROM test WHERE usernm = $myusername & password =$mypassword;

$result = MySQL_query($sql);

We are providing the prevention by using the methods which support by PHP php function mysql_real_escape_string to filter input data.As of PHP 5.5.0 mysql_real_escape_string and the mysql extension are deprecated.So we shall use mysqli extension and mysql_real_escape_string function instead.Where the unnecessary extra char-acters are removed and then passed
to the execution.

$myusername= stripslashes($myusername)

$mypassword= stripslashes($mypassword)

$myusername= mysql_ real_escape_string ($myusername)

$mypassword = mysql_real_escape_string ($mypassword)

We have implemented the following functions for effective Prevention of SQLi:

1. **Stripslashes() -** Un_quote string quoted with addcslashes().

2. **real_escape_string()** - Escapes special characters in a string for use in an Sql statement taking into account the current charset of the connection.

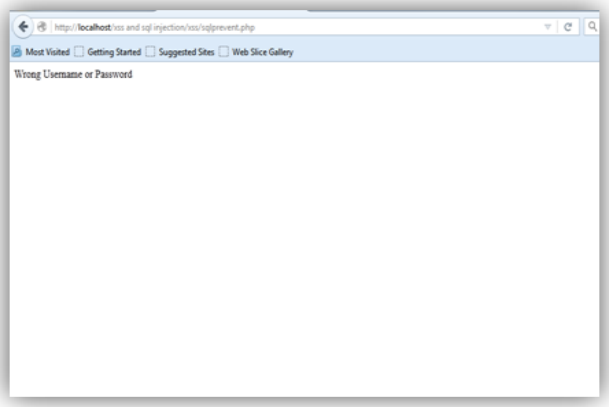


Fig-8: Working Of SQL Prevention

5.3 Prevention of Click-jacking Attack

The most widely used system for preventing clickjacking attack is No-Script (or NoScript Security Suite) which is a free and open-source extension for Mozilla Firefox, SeaMonkey, and other Mozilla-based web browsers. NoScript al-lows executable web content such as JavaScript, Java, Flash, Silverlight, and other plugins only if the site hosting is considered trusted by its user and has been pre-viously added to a whitelist. NoScript also offers specific countermeasures against security exploits. But in our system, we have disabled the iframes that result into the occurence of clickjacking attack.

iFrame is an important component while executing a click-jacking attack. i-frame is nothing but an area of the screen which hosts the infected code. The area can be effectively blocked to prevent this attack using the following snippet:

document.getElementById("blank").style.display="block";

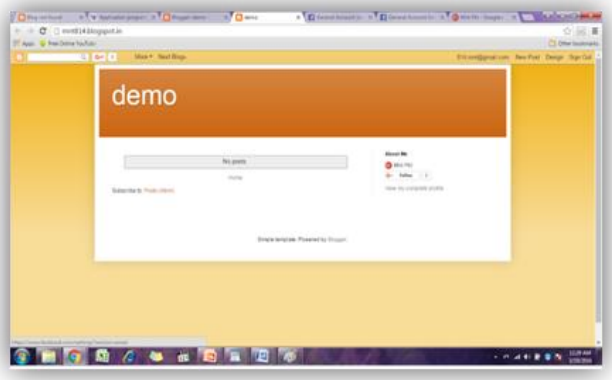


Fig-9: Hide iframes to Prevent Click-jacking Attack

6. CONCLUSIONS

This project was designed specially to cater to the growing demands of the security domain and focuses mainly on the website attacks and their protection. We have handled various attack modules in this project viz SQLi, Cross-site Scripting attacks (XSS), Click-jack and tab-nabbing attacks. Through this project, we have aimed to build a system that can successfully detect these attacks and prevent them automatically. Some browsers already provide protection against these attacks but our system is designed to run universally. It is applicable in almost every field of work be it banking or e-commerce and can also be essential for people who work from home.

The future research will be considerate to constructAPIs to detect SQLi and XSS web attacks.

REFERENCES

[1] Jose Fonseca, Nuno Seixas, Marco Vieira, and Henrique Madeira, Analysis
of Field Data on Web Security Vulnerabilities"in IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 11, NO. 2, MARCH/APRIL 2014.

[2] S.SHALINI, S.USHA, Prevention Of Cross-Site Scripting Attacks (XSS) On
Web Applications In The Client Side.

[3] Lwin Khin Shar , Hee Beng Kuan Tan, Automated removal of cross site scripting
vulnerabilities in web applications in 2011.

[4] William G.J. Halfond, Jeremy Viegas, and Alessandro Orso, "A Classification of SQL Injection Attack and Countermeasures.

[5] Kuldeep Kumar, Dr. Debasish Jena and Ravi Kumar "A Novel Approach to detect SQL injection in web applications"

[6] F. Valeur, D. Mutz, and G. Vigna. A learning-based approach to the detection of sql attacks. LNCS, 3548:123–140,2005.

[7] W.G. Halfond, J. Viegas, and A. Orso, "A Classification of SQLInjectionAttacks and Countermeasures," Proc. IEEE Int'l Symp.Secure Software Eng., Mar. 2006.

[8]http://www.slideshare.net/RespaPeter/types-of-sql-injection-attacks

[9]https://blogs.sophos.com/2013/11/01/how-malware-works-anatomy-of-an-attack-in-five-stages-infographic/

BIOGRAPHIES

Name: Pathak Mrunali P.
Qualification: BE[computer]

Name: Khan Nida Kausar I.
Qualification: BE[computer]

Name: Tantak Tejashree C.
Qualification: BE[computer]